

UNIVERSIDADE DE LISBOA



**Ensino de variáveis estruturadas String em linguagem C
com recurso à plataforma Arduino**

Paulo de Oliveira Ferreira

Mestrado em Ensino de Informática

**Relatório da prática de ensino supervisionada orientado pelo Professor Doutor
João Manuel Nunes Piedade e coorientado pela Professora Doutora Ana Paula
Pereira Afonso**

2020

Dedico à minha mãe!

Que num delicado equilíbrio de orgulho e humildade, onde quer que esteja me faz sempre saber “...de onde venho.”, e o significado de incondicionalidade.

Ao meu filho!

Que esteve presente no ato de formalização do reingresso neste “desafio”, e que nesta fase final, na sua suposta inocência me pergunta: “Já acabaste?...”. Sim, por ti e por todos como tu!...

Agradecimentos

Ao Professor João Piedade, pela sua grande capacidade de orientação e coerência. Porque em qualquer desafio é preciso que alguém acredite no objetivo para além de nós próprios, partilhando genuinamente a sua visão e profissionalismo.

À professora Ana Afonso, pela sua capacidade de perceção e orientações assertivas. Porque para todo o objetivo e inerentes obstáculos é necessário motivação e uma visão lucida das razões humanas.

À professora Cláudia Barata e aos seus alunos desta turma, pela colaboração e recetividade imprescindíveis e únicos, que no conjunto me inspiraram para transformar o esforço em determinação e realização.

Ao Instituto de Educação da Universidade de Lisboa, que no seu conjunto como escola e símbolo do ensino em Portugal proporciona a transformação da Investigação e Formação Educacional em experiências humanas ricas em valores e contributo para o desenvolvimento da nossa sociedade.

A todos os amigos que genuinamente partilharam comigo os momentos difíceis e os de inspiração neste percurso, assim como os que acreditaram, e de alguma forma concreta contribuíram para a possibilidade desta meta...

Resumo

Realizado no âmbito da unidade Curricular Iniciação à Prática Profissional IV do Mestrado em Ensino de Informática da Universidade de Lisboa, no Instituto de Educação, este relatório documenta a experiência vivida na Prática de Ensino Supervisionada (PES), realizada na Escola Secundária de Sacavém, do Agrupamento de Escolas Eduardo Gageiro. Esta intervenção pedagógica foi realizada numa turma do 1.º ano do Curso Profissional Técnico de Gestão e Programação de Sistemas Informáticos, na disciplina Programação e Sistemas de Informação, em particular no módulo 4 – Estruturas de Dados Estáticas.

Este módulo tem como temática principal as estruturas de dados estáticas na linguagem programação C, propriamente os Arrays nos quais se inserem as variáveis String. As atividades pedagógicas incidiram sobre os conceitos relacionados com a sua definição e manipulação.

Considerando a natureza dos conteúdos, e a problemática associada a este nível do ensino de programação, as atividades de aprendizagem planeadas foram organizadas segundo os princípios da metodologia de Aprendizagem Baseada em Problemas.

Assim, foram desenhados vários problemas que os alunos tiveram que desenvolver para aplicar e dominar os respetivos conteúdos. Na primeira parte da PES os exercícios foram realizados na IDE DEV C++, e na segunda parte o problema final pretendeu que os alunos implementassem um sistema cujo funcionamento principal consiste na visualização de Strings através de um display LCD modular ligado ao Microcontrolador Arduino. Essas Strings apresentam valores de temperatura e

humidade ambiente captados através de um módulo dedicado também ligado ao Arduino.

A análise dos resultados, relativa à evolução da aprendizagem dos alunos apresenta uma melhoria nos vários domínios de avaliação sumativa, entre a primeira e a segunda parte da PES, indicando também que esse fato esteve diretamente relacionado com o aumento do grau de interesse e satisfação com a utilização da plataforma Arduino nas atividades, assim como com a sua adequação à metodologia PBL e respetivas estratégias.

Palavras-chave: Linguagem C, Strings, Aprendizagem por problemas, DEV C++, Arduino.

Abstract

Within the scope of the Initiation of the Professional Practice IV curricular unit of the Master's degree in Informatics Teaching from the University of Lisbon, Institute of Education, this report describes the experience result from the Supervised Teaching Practice (STP), held at Sacavém High School, from the school group Eduardo Gageiro. This pedagogical intervention was realised in the 1st year class from the Professional Course of Computer Systems Management and Programming Technician, at the discipline of Programming and Information Systems, particularly in the Module 4 – Static Data structures.

This module has as main theme the static data structures in the C programming language, properly the Arrays in which the String variables are inserted. The pedagogical activities focused on the concepts related to its definition and manipulation.

Considering the nature of this theme contents, and the current issue related to teaching of this first programming level, the planned learning activities were organized and oriented by the Problem base learning methodology principles.

Thus, several problems were designed that the students had to develop in order to apply and master the respective contents. In the first part of the STP, the exercises were carried out in IDE DEV C ++, and in the second part the final problem was that the students implemented a system whose main function is the visualization of Strings through a modular LCD display connected to the Arduino Microcontroller. These Strings present ambient temperature and humidity values captured through a dedicated module also connected to the Arduino.

The analysis of the results, related to the evolution of the students' learning, shows an improvement in the several domains of summative assessment, between the first and the second part of the STP, also indicating that this fact was directly related to the increase in the degree of interest and satisfaction with the use of the Arduino platform in activities, as well as its adaptation to the PBL methodology and respective strategies.

Keywords: C language, Strings, Problem Based Learning, DEV C ++, Arduino.

Índice

Dedicatória	i
Agradecimentos.....	ii
Resumo.....	iii
Abstract	v
Índice de figuras.....	xi
Índice de tabelas	xiii
1. Introdução	1
2. Contexto da Intervenção	4
2.1. O agrupamento de escolas Eduardo Gageiro	4
2.1.1. A escola secundária de Sacavém.....	7
2.1.2. Caracterização da turma	10
3. Enquadramento curricular e didático	15
3.1. O curso profissional de Técnico de Gestão e Programação de Sistemas Informáticos.....	16
3.1.1. A disciplina de Programação e Sistemas de Informação	18
3.2. Módulo 4 – Estruturas de Dados Estáticas	20
3.3. Análise crítica do programa da disciplina	22
3.4. Problemática e estado atual no Ensino da Programação	26
4. Intervenção Pedagógica	31
4.1. Fundamentação e enquadramento conceptual	31

4.1.1.	Definição de String	34
4.1.2.	Caracteres e Strings.....	36
4.1.3.	Declaração e Inicialização.....	38
4.1.4.	Strings e ponteiros para Strings.....	42
4.2.	Objetivos da lecionação e aprendizagem na PES	45
4.3.	Metodologias de ensino	47
4.4.	Planificação	49
4.4.1.	Sobre os principais recursos utilizados	50
4.4.2.	Estratégias	52
4.5.	Descrição da operacionalização das aulas	58
4.5.1.	Descrição das aulas 1 a 3 – Primeiro dia.....	58
4.5.2.	Descrição das aulas 4 a 7 - Segundo dia	63
4.5.3.	Descrição das aulas 8 a 10 – Terceiro dia	70
4.5.4.	Descrição das aulas 11 a 14 – Quarto dia	73
5.	Avaliação da aprendizagem	77
5.1.	Metodologia e instrumentos de avaliação das aprendizagens	77
5.2.	Avaliação diagnóstica.....	79
5.3.	Avaliação formativa	81
5.4.	Avaliação sumativa	82
6.	Componente investigativa da Intervenção	87
6.1.	Dimensão Investigativa «Metodologia».....	91
6.2.	Dimensão Investigativa «Recursos Pedagógicos»	99

6.2.1.	Características específicas do Ambiente de Desenvolvimento.....	99
6.2.2.	Dificuldades nos conceitos e percepção de utilidade e interesse	101
6.2.3.	Percepção de utilidade das Strings em aplicações reais.....	102
6.2.4.	Cativação e interesse nos dois Ambientes de Desenvolvimento	103
6.3.	Análise global.....	104
7.	Reflexão final.....	106
	Referências.....	111
	Legislação Consultada	114
	Anexos	i
	Anexo A - Teste diagnóstico.....	iii
	Anexo A1 – Correção do teste diagnóstico.....	vi
	Anexo B – Apresentação Expositiva de conceitos em PowerPoint	vii
	Anexo C – Ficha de exercícios em DEV C++	xii
	Anexo D - Teste de Aferição	xxiii
	Anexo E1 – Planificações das aulas do dia 3/03/20.....	i
	Anexo E2 – Planificações das aulas do dia 05/03/20.....	iii
	Anexo E3 – Planificações das aulas do dia 10/03/20.....	v
	Anexo E4 – Planificações das aulas do dia 12/03/20.....	vi
	Anexo F - Respostas aos questionários investigativos.....	i
	Anexo G1 – Avaliação sumativa – Dimensão Atitudes e valores no DEV C++	i
	Anexo G2 – Avaliação sumativa – Dimensão Atitudes e valores no Arduino	iv

Anexo G3 – Avaliação sumativa – Dimensão Conhecimento e capacidades cognitivas DEV C++	v
Anexo G4 – Avaliação sumativa – Dimensão Conhecimento e capacidades cognitivas Arduino	vi
Anexo G5 – Avaliação sumativa – Dimensão Conhecimento e capacidades cognitivas Teste Final	xi
Anexo G6 – Avaliação sumativa – Resultado Global.....	xii
Anexo H – Manual de conceitos teóricos	i
ANEXO I – Manual e Guião da atividade Ardumotic	ix
ANEXO J – Tabela ASCII.....	xxii

Índice de figuras

Figura 1 - <i>Agrupamento de Escolas Eduardo Gageiro</i>	6
Figura 2 - <i>Escola Secundária de Sacavém</i>	8
Figura 3 - Posição habitual dos alunos na sala de aula.	11
Figura 4 - Diagrama de conceitos global da temática Strings.....	33
Figura 5 - Representação da estrutura de uma String	36
Figura 6 - Conjunto Arduino e Módulos, Display I2C e DHT11, com respetiva ligação.	57
Figura 7 - PrintScreen do aluno D demonstrativo do exercício 5.a).....	66
Figura 8 - <i>PrintScreen do aluno D demonstrativo do exercício 6.a)</i>	67
Figura 9 - <i>PrintScreen do aluno D demonstrativo do exercício 7.</i>	68
Figura 10 - <i>Exemplo de montagem e funcionamento do grupo V, com display nas duas linhas independentes, relacionado com o exercício do ponto 4</i>	72
Figura 11 - <i>Exemplo de implementação do grupo VI.</i>	76
Figura 12 - <i>Evolução global da aprendizagem no conjunto das dimensões constituintes das Aprendizagens Essenciais.</i>	86
Figura 13 - <i>Evolução da componente Atitudes e Valores das AE</i>	88
Figura 14 - <i>Evolução dos Conhecimentos e Capacidades Cognitivas das AE.</i>	88
Figura 15 - Representação por blocos da Componente Investigativa da PES	90
Figura 16 – <i>Representação da Metodologia de Aprendizagem por Problemas, modelo de Gick (1986)</i>	91
Figura 17 – <i>Resposta à questão 3 do questionário 1, no final da primeira parte da PES</i>	94
Figura 18 - <i>Respostas à questão 3, no final da segunda parte da Intervenção</i>	94

Figura 19 - <i>Respostas dos alunos ao desempenho do Professor nas duas partes da PES. Escala de resposta: 1-Discordo; 2-Neutro; 3-Concordo; 4 – Concordo plenamente.</i>	96
Figura 20 - <i>Média das respostas obtidas sobre o grau de dificuldade na respectiva IDE de desenvolvimento. Escala de resposta: (Escala: 1-pouca; 5-muita).</i>	100
Figura 21 – <i>Nível e evolução da percepção de utilidade, interesse e dificuldade na compreensão de conceitos. Escala de resposta: 1-Nenhuma; 5-Muita</i>	101
Figura 22 - <i>Percepção de Utilidade das Strings em Aplicações reais. Escala: 1 – Nenhuma; 4- Muita.</i>	102
Figura 23 – <i>Respostas sobre o interesse na aprendizagem sobre Strings, na IDE DEV C++ e Arduino respectivamente.</i>	103

Índice de tabelas

Tabela 1 - <i>Resultados de avaliação sumativa dos alunos nos módulos anteriores.</i> ..	14
Tabela 2 - Estrutura do Curso Gestão e Programação de Sistemas Informáticos.....	17
Tabela 3 - Constituição modular da disciplina PSI.....	19
Tabela 4 - Módulos opcionais da disciplina PSI (duração de referência).....	20
Tabela 5 - Representação em memória de uma String e uma variável char.	37
Tabela 6 - Representação em memória de um ponteiro para String	42
Tabela 7 - <i>Representação em memória de uma String, um ponteiro para String e uma atribuição do endereço da String ao ponteiro.</i>	43
Tabela 8 - Representação em memória de uma String e um ponteiro para String.....	44
Tabela 9 - Correspondência de ligações entre o Arduino e o Display I2C	58
Tabela 10 – <i>Observações dos resultados da avaliação diagnóstica.</i>	80
Tabela 11 - <i>Componentes da avaliação sumativa baseada nas ACPA, com respetivos domínios, ponderação e elementos.</i>	83
Tabela 12 - <i>Classificação Global dos alunos por componentes de avaliação, das duas partes da PES.</i>	85
Tabela 13 - Grupos de respostas por tipo e com respetivo peso na opinião global sobre as aulas nas duas partes da PES.....	95

1. Introdução

No âmbito da unidade curricular Iniciação à Prática Profissional IV do Mestrado em Ensino de Informática da Universidade de Lisboa, no Instituto de Educação, este relatório documenta e descreve o Projeto da Intervenção Pedagógica (Também designada abreviadamente por Intervenção) da Prática de Ensino Supervisionada (PES), realizada na Escola Secundária de Sacavém, do Agrupamento de Escolas Eduardo Gageiro. A Intervenção foi realizada com a turma do Curso Profissional Técnico de Gestão e Programação de Sistemas Informáticos, na disciplina Programação e Sistemas de Informação, no Módulo 4 – Estruturas de dados estáticos, na temática Strings. A duração total da Intervenção foi de 14 aulas de 45min realizadas entre o dia 3 e 12 de março do ano 2020.

A preparação de aulas desta PES foi feita com o indispensável apoio dos professores orientadores e da professora cooperante, que estiveram sempre disponíveis para contribuir nas várias fases deste trabalho.

Tendo em conta a problemática relacionada com o ensino da programação a este nível introdutório, definiu-se como estratégia principal a possibilidade de ligar as aprendizagens a uma aplicação final real, de modo a tirar partido dos pontos fortes da metodologia PBL (Problem Based Learning) adotada, assim como estimular a motivação e autonomia dos alunos para além do simples resultado da avaliação sumativa.

Como ponto de partida da estratégia principal desta PES optou-se por utilizar 2 recursos educativos diferentes em duas fases respetivamente. Esta opção teve como objetivo permitir uma abordagem inicial aos conceitos em ambiente de trabalho familiar, através da IDE DEV C++ na primeira parte, e na segunda parte implementar

um pequeno projeto guiado na plataforma Arduino, contendo alguns problemas para resolver aplicando os conhecimentos adquiridos.

Com esta estratégia, e para a investigação e análise final aqui pretendida, não é aqui o propósito comparar os dois recursos educativos nos seus critérios pedagógicos mas sim analisar a evolução da aprendizagem dos alunos em algumas dimensões pedagógicas à medida que os mesmos foram sendo utilizados. Pretende-se que a utilização de dois diferentes recursos seja complementar, e proporcione aos alunos uma visão holística da forma de aplicação de conceitos e da sua utilidade, com ênfase na experiência da sua aplicação real. Desta forma um dos aspetos principais aqui considerado é também a percepção manifestada pelos alunos em relação a esta metodologia adotada e à experiência vivida nestas atividades propostas.

O relatório aqui apresentado é constituído por 7 capítulos: Na introdução apresenta-se o trabalho na sua estrutura global e finalidades; O segundo capítulo caracteriza o contexto da Intervenção, o local onde decorre, a respetiva escola, a sala de aulas e a turma; O terceiro capítulo faz um enquadramento curricular e didático, descrevendo de forma geral o Curso Profissional em questão, a disciplina e módulo de incidência da Intervenção. Apresenta também uma análise crítica ao programa, assim como uma abordagem à problemática e estado atual do ensino desta disciplina. O quarto capítulo apresenta o plano concreto desta Intervenção, partindo da descrição detalhada dos conceitos envolvidos e a sua fundamentação. Faz uma descrição da principal metodologia adotada e da sua aplicação concreta em estratégias e recursos. Deste conjunto resultam os planos de aulas, cuja operacionalização é também descrita. No quinto capítulo apresenta-se a avaliação das aprendizagens nas várias dimensões, assim como a avaliação das aulas na perspetiva dos alunos. No sexto capítulo é apresentada a componente investigativa da PES, assente em questões sobre a relação

entre resultados obtidos na aprendizagem com alguns fatores concretos, permitindo extrair informação útil e conclusiva da sua análise. No sétimo e último capítulo é feita uma reflexão global de todo o processo desta PES, numa perspetiva pessoal e profissional.

2. Contexto da Intervenção

A Intervenção pedagógica aqui exposta parte da percepção de uma realidade específica de um contexto educativo integrado na nossa sociedade atual. Numa era de grandes organizações e globalização, está em causa o potencial criativo e a autonomia individual. Na perspetiva de Petitat (2006), relacionando a história, pedagogia e sociedade existe um ponto de vista construtivo que resulta da posição ativa do professor na contribuição da escola como produtor da sociedade.

Nesta abordagem inicial descreve-se de seguida a realidade atual deste contexto educativo, abrangendo desde o agrupamento de escolas até à caracterização dos alunos na turma, na dimensão que os caracteriza como cidadãos e como iniciantes de um Curso de Informática em que a programação é fundamental e cada vez mais transversal.

Para isso foram analisados os documentos que caracterizam o agrupamento de Escolas Eduardo Gageiro (Projeto Educativo, Plano Anual de Atividades e Regulamento Interno) e grelhas de caracterização da turma, cordialmente cedidas pela professora cooperante. Além da informação documental publicamente acessível, foram também muito importantes as visitas regulares às aulas da turma no horário da disciplina, antes de Intervenção, permitindo observar diretamente a dinâmica comportamental e escolar de cada aluno.

2.1. O agrupamento de escolas Eduardo Gageiro

Segundo informação disponibilizada pela escola nos seus documentos estruturantes, o Agrupamento de Escolas Eduardo Gageiro (denominação homologada a 18 de abril de 2013) nasceu da fusão do Agrupamento de Escolas de Sacavém e Prior Velho com a Escola Secundária de Sacavém (Sede), passando a ser constituído por

todos os estabelecimentos de educação e ensino (educação pré-escolar, ensino básico e ensino secundário) da área correspondente à união das freguesias de Sacavém e Prior Velho, na zona oriental do concelho de Loures.

Sendo inicialmente constituído por oito escolas, o Agrupamento contempla atualmente (e desde um de Setembro de 2013) sete estabelecimentos, por desativação da EB1 n.º 2 de Sacavém - há muitos anos instalada num edifício de habitação com evidentes limitações espaciais e com alguns problemas de segurança - conforme previsto na Carta Educativa Municipal de 2007.

A escola EB1 n.º 2 de Sacavém foi integrada na escola básica Bartolomeu Dias, passando esta a contemplar o 1º e 2º ciclos. Assim, o Agrupamento integra atualmente os seguintes estabelecimentos escolares, que se encontram representados geograficamente na figura 1:

(A) Escola Secundária de Sacavém (Escola Sede – 3º ciclo e Ensino Secundário)

(B +E) Escola Básica Bartolomeu Dias (1º e 2º ciclos)

(C) Escola EB do Prior Velho (com Jardim de Infância)

(D) Jardim de Infância da Quinta de São José - Sacavém

(F) Escola EB no 3 de Sacavém (com Jardim de Infância)

(G) Escola EB de Sacavém (com Jardim de Infância)

(H) Jardim de Infância Terraços da Ponte

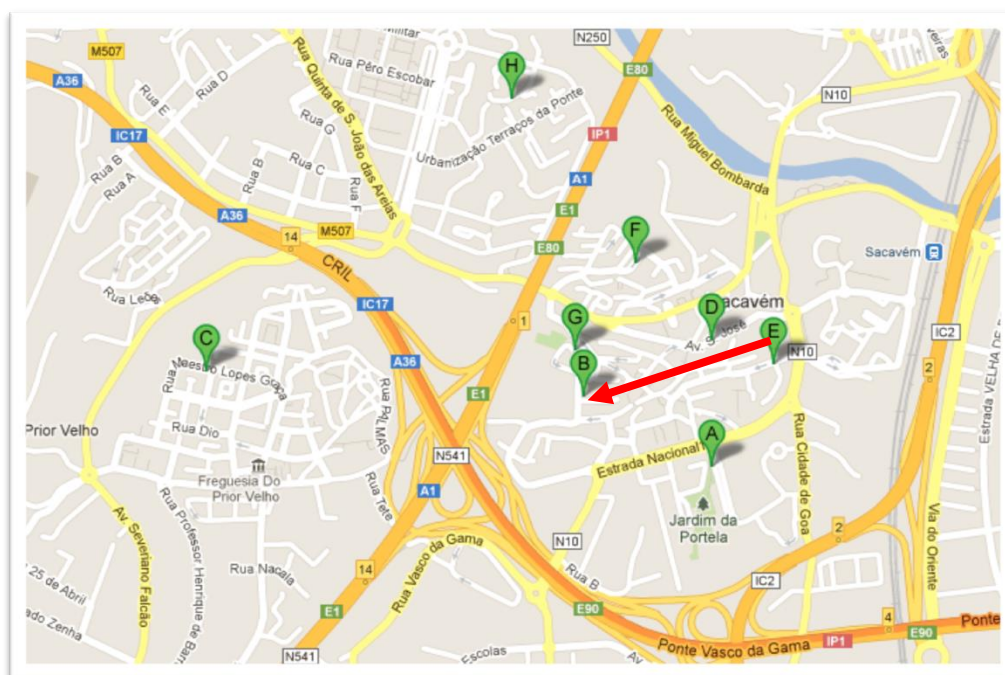


Figura 1 - Agrupamento de Escolas Eduardo Gageiro

No seu projeto educativo (2015-2019) é referido que a população escolar tem oscilado entre os 2300 e 2400 alunos que se distribuem pelos diferentes níveis de ensino, desde o pré-escolar à formação de adultos. A população discente do agrupamento é originária das freguesias e localidades circundantes, sendo que um número elevado de alunos têm nacionalidade estrangeira, e são oriundos dos sistemas educativos dos respetivos países. Conclui-se deste modo que a população é bastante diversificada em termos sociais, económicos e culturais. Ainda no mesmo documento são relatados os principais problemas com que o agrupamento se tem deparado, nomeadamente em relação ao abandono e insucesso escolar e algumas situações de indisciplina, sendo que estes indicadores têm vindo a melhorar significativamente nos últimos anos. Dadas as suas características, o Agrupamento de Escolas Eduardo Gageiro integra, desde há vários anos, o Programa TEIP (Territórios Educativos de Intervenção Prioritária), o que tem permitido desenhar propostas e planos educativos para responder aos principais problemas da comunidade educativa.

Este Programa foi criado em 1996 da iniciativa governamental, e na base de uma política de discriminação positiva e privilegiando a luta contra o insucesso escolar. Foi inspirado na política equivalente das ZEP (Zones d'Éducation Prioritaires) em França, e nas experiências de educação compensatória desenvolvidas nos EUA, no quadro dos programas de “guerra à pobreza” e nas experiências britânicas da mesma época (Ferreira & Teixeira, 2010).

O Programa TEIP é implementado atualmente em 137 agrupamentos de escolas e escolas não agrupadas que se localizam em territórios económica e socialmente desfavorecidos, marcados pela pobreza e exclusão social, onde a violência, a indisciplina, o abandono e o insucesso escolar mais se manifestam (Direção-Geral da Educação [DGE], 2020). Segundo a DGE, os principais objetivos deste programa são a prevenção e redução do abandono escolar precoce e do absentismo, a redução da indisciplina e a promoção do sucesso educativo de todos os alunos (DGE, 2020).

2.1.1. A escola secundária de Sacavém

A escola Secundária de Sacavém, foi criada pela Portaria n.º 244/77, de 9 de maio e funcionou de forma provisória na Portela de Sacavém entre 1976 e 1982. A partir desse ano passou a funcionar na cidade de Sacavém nas atuais instalações (Figura 2). No âmbito da reorganização da rede escolar promovida pela Resolução do Conselho de Ministros n.º 44/2010, a escola passou a integrar o Agrupamento de Escolas, assumindo-se como escola sede.

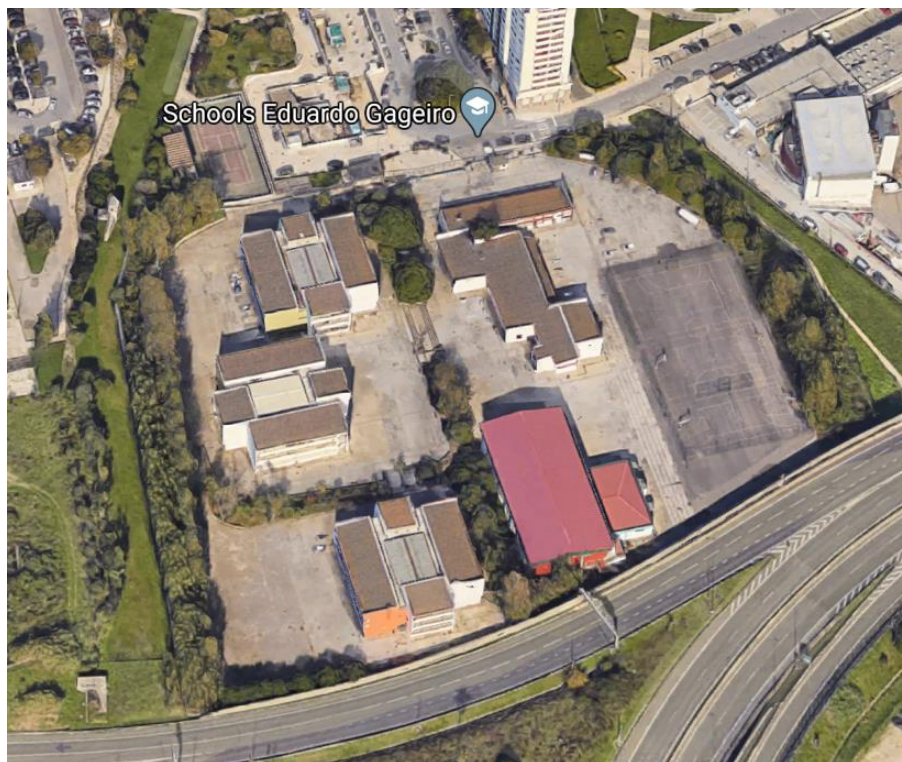


Figura 2 - *Escola Secundária de Sacavém.*

No Regulamento Interno do agrupamento é possível perceber que a escola sede dispõe de um conjunto significativo de espaços destinados a atividades letivas, constituídos por salas de aula, salas e laboratórios específicos e espaços diversificados para trabalho dos professores. Em termos tecnológicos a escola foi intervencionada em 2007 no âmbito do Plano Tecnológico da Educação e encontra-se equipada com 1 computador e videoprojector por sala, e vários laboratórios de informática. Apesar de grande parte do parque informático da escola ter cerca de 13 anos, tem existido nos últimos anos a modernização de alguns dos laboratórios com materiais mais recentes. A escola dispõe ainda de alguns recursos tecnológicos, como kits robóticos, drones, kits Arduino que permitem a realização de projetos nas disciplinas de Informática presentes nos diferentes cursos da oferta formativa. Assim, considera-se que esta escola tem equipamento adequado para as suas atividades letivas, e o suporte de uma equipa TIC que contribui para o funcionamento e manutenção destas tecnologias, em

Hardware e Software, nas várias dimensões, desde a segurança aos projetos e atividades pedagógicas.

O projeto educativo do agrupamento (2015-2019) define um conjunto de objetivos e ações estratégicas que visam a melhoria das aprendizagens dos alunos, tendo como meta que os alunos “aprendam a ser cidadãos do mundo” (Agrupamento de Escolas Eduardo Gageiro [AEEG], 2015). Assim, a escola dispõe de uma vasta oferta formativa, que procura dar resposta aos objetivos definidos e aos problemas identificados.

A oferta formativa da escola secundária de Sacavém é a seguinte:

- 3º Ciclo;
- Cursos de Educação e formação;
- Cursos científico-humanísticos:
 - Curso de Ciências e Tecnologias;
 - Curso de Ciências Socioeconómicas;
 - Curso de Línguas e Humanidades.
- Cursos profissionais:
 - Técnico Comercial
 - Técnico de Design Gráfico
 - Técnico de Gestão e Programação de Sistemas informáticos
 - Técnico de instalações elétricas
 - Técnico de turismo
- Ensino noturno;
- Português para falantes de outras línguas (PFOL)

Além da oferta formativa, a escola dispõe de várias atividades extracurriculares que contribuem para a formação integral dos seus alunos. Destacam-se os clubes e projetos:

- RoboPess - Clube de Informática, Programação e Robótica;
- Clube de ciência;
- Oficina das artes;
- Projeto Pequeno cientista;
- Projeto Apps for Good;
- Projeto Erasmus + New Dimensions of Education;
- Projeto Ser Pro;
- Orquestra Bora Nessa;
- Concurso “Cria um site”, para os alunos dos cursos profissionais e disciplinas de TIC;
- LAN Party – Torneio de jogos e Playstation;
- Semana Europeia da Programação – seminários e workshops de Programação.

2.1.2. Caracterização da turma

A turma do curso profissional de gestão e programação de sistemas informáticos (GPSI), onde decorreu a Intervenção, ate à data de início da Intervenção é constituída por 14 alunos, 12 rapazes e 2 raparigas. Os alunos da turma têm idades compreendidas entre 14 e 17 anos, oito alunos apresentam pelo menos uma reprovação no seu percurso escolar e seis alunos nunca reprovaram. A maioria dos alunos tem computador em casa (12), indica que já programou dispositivos móveis (12) e consideram os seus conhecimentos prévios em programação razoáveis (3 numa escala

de 5). De referir que 3 dos alunos consideram os seus conhecimentos prévios bons e 2 consideram fracos os seus conhecimentos de programação.

A sala de aulas utilizada para a disciplina PSI é a sala C218, sendo a mais completa em termos de equipamento informático, e encontra-se próxima da sala de armazenamento de recursos informáticos e eletrónicos.

Nesta sala os alunos distribuem-se de acordo com a representação da figura 3. Quando existem demonstrações feitas pelo professor ou seja necessário um espaço comum são utilizadas as 4 mesas ao meio da sala. Quando existem atividades de grupo, normalmente mantêm-se nas mesas periféricas onde existem os computadores.

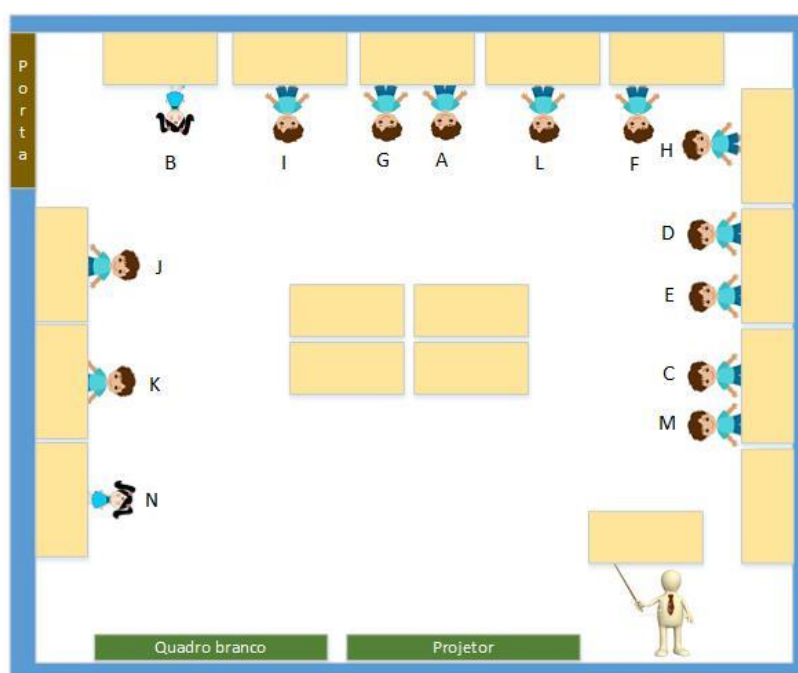


Figura 3 - *Posição habitual dos alunos na sala de aula.*

As aulas da disciplina Programação e Sistemas de Informação (PSI) decorrem distribuídas em dois dias da semana, de acordo com o horário da figura seguinte, correspondendo a 3 tempos letivos de 45 minutos à 3ª feira e 4 tempos letivos de 45 minutos à 5ª feira.

A turma do curso profissional de programação de sistemas informáticos (GPSI) e a do curso profissional Técnico de Instalações Elétricas (ELEL) têm em comum as disciplinas da componente Sociocultural e Científica, separando-se nas disciplinas da componente técnica, como representado na figura seguinte.

1	8:15 9:00			ELEL IE B336 RC. GPSI C324	DESQ IE B336 RC. GPSI C324	PORT A110
2	9:00 9:45	AI C219	FQU T1 A110			
3	10:00 10:45		ELEL IE B336 PRSI. GPSI C218	TA IE C213 SO. GPSI C324		
4	10:45 11:30	LE-ING B105			ELEL IE B336 PRSI. GPSI C218	MAT A110
5	11:45 12:30		PO IE B336 PRSI. GPSI C218			
6	12:30 13:15		PO IE B336	MAT A110		TIC C324
7	13:30 14:15	FQU A231				
8	14:15 15:00		FQU T2 A231		AI A221	FQU A231
9	15:15 16:00	PORT A110	LE-ING C217		EDF G1	
10	16:00 16:45					
11	17:00 17:45	TA IE C213 SO. GPSI C324	TIC C324		PORT A110	PO IE B336 AC. GPSI C324
12	17:45 18:30					

Quadro 1 - Horário da turma GPSI

Da observação das várias aulas assistidas antes da PES, constatei que a maioria dos alunos demonstra estar satisfeito com a escolha deste curso. Para indagar as suas expetativas e visão das aulas de programação, numa das aulas assistidas foi-lhes perguntado qual o tipo de atividades práticas que gostavam de ter, e a resposta quase unânime foi atividades de “programação com drones” (sabendo eles à partida que esses recursos existem na escola). Possivelmente a sua perspetiva (Sem pretender classificar, pois isso refere-se a metodologias de ensino) talvez fosse ver os Drones antes de mais como um brinquedo, o que não deixou de tornar a resposta interessante.

Em algumas observações de outras aulas, durante a resolução de exercícios de programação usando o ambiente de desenvolvimento integrado (IDE) DEV C++ fui interagindo com os alunos tentando aperceber-me do grau de assimilação de conceitos

básicos, como por exemplo tipos de variáveis, constatando que efetivamente no início da aprendizagem nesta disciplina, até a distinção entre um número inteiro e um número real pode ser complexo. No entanto, pelo menos três alunos já apresentavam bons conhecimentos na utilização do computador como ferramenta de trabalho e desenvolvimento.

De forma sintética e numa descrição da experiência com os alunos nesta fase preliminar, poderia apresentar os seguintes descritores com pequenos episódios de interação:

- São calmos, normalmente cooperantes e disciplinados, com apenas 2 ou três que se envolvem regularmente em brincadeiras ou em assuntos de outros interesses que não da disciplina, o que por vezes leva o professor a interromper a aula;
- São bastante empenhados em atividades práticas. Na última aula assistida, tive oportunidade de partilhar com eles a atividade de programação do Drone Parrot Mambo. O seu envolvimento foi sempre ordeiro, empenhado e consciente da organização e manutenção do equipamento. Apesar de num dos grupos ter havido algum desentendimento entre dois alunos, simplesmente pela concorrência na programação do Drone, mas acabaram por ficar mais calmos e entendidos. Pude ainda partilhar com aluno B a dificuldade de implementação de dois ciclos “for” aninhados. Em várias tentativas o resultado não era o esperado. Ajudei o aluno no seu raciocínio, confirmando que estava correto, assim como a sua implementação;
- São persistentes na procura de soluções para os exercícios, embora os métodos que utilizam sejam tendencialmente através do improviso e não pela procura das bases dos conceitos.

Relativamente à avaliação sumativa dos alunos até à fase da Intervenção, os resultados sumativos dos alunos, dos 3 primeiros módulos designadamente M1, M2 e M3, foram os seguintes:

Tabela 1 - *Resultados de avaliação sumativa dos alunos nos módulos anteriores.*

Aluno/ módulo	M1	M2	M3
I	17	18	18
H	18	18	17
C	17	18	17
D	17	16	16
A	14	13	16
G	17	17	16
B	14	15	16
M	17	17	16
K	17	15	15
J	13	12	13
E	14	13	13
F	13	13	13
L	12	10	10
Media	15	15	15

Pode constatar-se que não existem módulos em atraso. As médias da turma são de 15 valores nos três módulos, embora as notas mais baixas sejam respetivamente 12, 10 e 10 em cada módulo. Assim, estes alunos já trabalharam e adquiriam previamente os conceitos introdutórios de programação como algoritmia, ciclos de controlos de execução e Programação Estruturada através de Funções.

3. Enquadramento curricular e didático

Segundo o Decreto Lei nº55/2018, é necessário desenvolver nos alunos competências que lhes permitam questionar os saberes estabelecidos, integrar conhecimentos emergentes, comunicar eficientemente e resolver problemas complexos.

Pelo Despacho n.º 6944-A/2018 sobre a definição do Perfil dos Alunos à Saída da Escolaridade Obrigatória, considera-se que a promoção de um ensino de qualidade implica fomentar aprendizagens efetivas e significativas, com conhecimentos consolidados, que são mobilizados em situações concretas, favorecendo o desenvolvimento de competências de nível elevado.

Nesse sentido pretende-se usar como referência o Perfil dos Alunos à Saída da Escolaridade Obrigatória (DGE, 2017), que estabelece a matriz de princípios, valores e áreas de competências a que deve obedecer o desenvolvimento do currículo.

Filtrando o que a esta PES se aplica e considerando o efeito destas diretivas nas opções pedagógicas e métodos de avaliação aqui aplicados, pretende-se aplicar metodologias para promover as competências previstas no Perfil dos Alunos (PA) da seguinte forma:

- Abordar os conteúdos temáticos associando-os a situações e problemas presentes no quotidiano da vida do aluno ou presentes no meio sociocultural, recorrendo a materiais e recursos diversificados;
- Organizar as atividades letivas prevendo a experimentação de técnicas, instrumentos e formas de trabalho diversificados, promovendo intencionalmente na sala de aula atividades de observação, questionamento da realidade e integração de saberes;

- Organizar e desenvolver atividades cooperativas de aprendizagem, concretamente através de trabalhos de grupo, promovendo a partilha de conhecimento e respeito pelo próximo;
- Promover de modo sistemático e intencional, na sala de aula, atividades que permitam ao aluno fazer escolhas, confrontar pontos de vista, resolver problemas e estimular a criatividade.

3.1. O curso profissional de Técnico de Gestão e Programação de Sistemas Informáticos

Em 2004, no seguimento do Decreto-lei nº74/2004, de 26 de Março, definiram-se os princípios da revisão curricular do ensino profissional e da racionalização da sua oferta formativa, privilegiando a inserção dos alunos no mundo de trabalho e permitindo também o prosseguimento de estudos. Subsequentemente, pela Portaria n.º 550-C/2004, de 21 de Maio procedeu-se à reestruturação dos cursos em vigor, e à aprovação dos novos cursos e planos de estudos, à luz das novas regras e matriz curricular estabelecida pelos princípios do referido Decreto.

Resultando das diretivas atrás referidas, pela portaria 916/2005 do Ministério da educação, foi criado o Curso Profissional de Técnico de Gestão e Programação de Sistemas Informáticos, resultando da reestruturação do Curso Profissional Técnico de Gestão de Sistemas Informáticos, o qual terá sido extinto nesta data (Exceto para os alunos ainda a finalizá-lo).

Este novo curso profissional da categoria de Informática e da área de Educação e Formação 481 - Ciências Informáticas - confere aos alunos, além do diploma de conclusão do nível secundário de educação, também um certificado de qualificação profissional de nível IV, com o qual deverão estar aptos a realizar, de forma autónoma

ou integrado numa equipa, atividades de conceção, especificação, projeto, implementação, avaliação, suporte e manutenção de sistemas informáticos e de tecnologias de processamento e transmissão de dados e informações.

Atualmente, e após algumas adaptações, o Curso Profissional de Técnico de Gestão e Programação de Sistemas Informáticos tem a seguinte constituição, encontrando-se destacada na tabela a disciplina desta Intervenção:

Tabela 2 - Estrutura do Curso Gestão e Programação de Sistemas Informáticos

Componentes da formação / disciplinas		1.º ano	2.º ano	3.º ano	Total de horas
		Horas			
F. Sociocultural	Português	126	112	82	320
	Língua Estrangeira	99	72	49	220
	Área de Integração	74	73	73	220
	Educação Física	51	39	50	140
	TIC	100	---	---	100
	Subtotal	450	296	254	1000
F. Científica	Física e Química	100	100	---	200
	Matemática	99	99	102	300
	Subtotal	199	199	102	500
F. Técnica	Arquitetura de Computadores	72	80	---	152
	Sistemas Operativos	87	57	---	144
	Redes de Comunicação	102	90	60	252
	Programação e Sist. de Inf.	156	156	240	552
	Subtotal	417	383	300	1100
	PAP	---	---	---	---
	FCT	---	220	380	600
Total		1066	1098	1036	3200

Este curso é constituído pelas componentes de Formação Sociocultural, Científica e Técnica, sendo que desta última faz parte a disciplina Programação e Sistemas de Informação.

O total de carga horária da componente de formação técnica, 1100 horas, foi definido pela escola ao abrigo das alterações do Decreto-Lei n.º 55/2018 de 6 de julho, anexo VIII, pelo qual o total da carga horária da componente de formação técnica passou a estar entre 1000 e 1300. Desta forma resultou também a redefinição da carga horária da disciplina PSI, de 632 horas para 552 no total dos 3 anos, e sendo que 90 horas são dos módulos opcionais.

3.1.1. A disciplina de Programação e Sistemas de Informação

A disciplina de Programação e Sistemas de Informação, como vimos atrás, integra-se na componente técnica do curso, e a sua estrutura resulta da junção dos programas que existiam no curso anterior das disciplinas de Programação e Base de Dados. Segundo o programa inicialmente definido pela Direção Geral de Formação Vocacional (DGFV) (2005) para esta disciplina as finalidades são as seguintes:

- Fomentar a disponibilidade para uma aprendizagem ao longo da vida como condição necessária à adaptação a novas situações e à capacidade de resolver problemas no contexto da sociedade do conhecimento;
- Promover a autonomia, a criatividade, a responsabilidade, bem como a capacidade para trabalhar em equipa numa perspetiva de abertura à mudança, à diversidade cultural e ao exercício de uma cidadania ativa;
- Fomentar o interesse pela pesquisa, pela descoberta e pela inovação, face aos desafios da sociedade do conhecimento;
- Fomentar competências no planeamento e apresentação de soluções informáticas;
- Promover as práticas de diagnóstico e a correção de falhas no funcionamento de sistemas informáticos;

- Desenvolver a capacidade de analisar de forma objetiva as linguagens de programação existentes;
- Fomentar a capacidade de compreender as técnicas básicas de implementação de linguagens de programação, e desenvolver uma capacidade acrescida de aprender novas linguagens de programação, assim como uma acrescida capacidade de conceção e desenvolvimento de software e sistemas de informação;

A constituição modular da disciplina está representada na tabela 3, sendo que os três últimos módulos são de escolha opcional de entre a lista de 7 disponíveis na tabela 4, resultando num total de 552 horas letivas. Os docentes podem seleccionar de entre os módulos opcionais aqueles que consideram mais adequados para complementar a formação dos seus alunos.

**Tabela 3 - Constituição modular da disciplina
DCI**

Número	Designação (obrigatórios)	Duração de referência (horas)
1	Introdução à Programação e Algoritmia	30
2	Mecanismos de Controlo de Execução	30
3	Programação Estruturada	36
4	Estruturas de Dados Estáticas	30
5	Estruturas de Dados Compostas	30
6	Estruturas de Dados Dinâmicas	36
7	Tratamento de Ficheiros	30
8	Conceitos Avançados de Programação	18
9	Introdução à Programação Orientada a Objectos	36
10	Programação Orientada a Objectos	36
11	Programação Orientada a Objectos Avançada	30
12	Introdução aos Sistemas de Informação	21
13	Técnicas de Modelação de Dados	19
14	Linguagem de Manipulação de Dados	19
15	Linguagem de Definição de Dados	21
16	Projecto de Software	50
17 (t)	Tema opcional	30
18 (t)	Tema opcional	30
19 (t)	Tema opcional	20

Tabela 4 - *Módulos opcionais da disciplina PSI (duração de referência).*

Numero	Designação	Duração de referência (horas)
OP1	Tecnologias de Acesso a Bases de Dados	30
OP 2	Técnicas de Detecção e Tratamento de Erros	30
OP 3	Metodologias de Análise e Desenvolvimento de Sistemas	30
OP 4	Conceitos de Organização e Gestão de Empresas	30
OP 5	Ferramentas de Desenvolvimento de Páginas Web	30
OP 6	Ferramentas de Animação Gráfica	30
OP 7	Ferramentas de Tratamento de Imagem	30

3.2. Módulo 4 – Estruturas de Dados Estáticas

Esta Intervenção incide no Módulo 4 da disciplina PSI - Estruturas de Dados Estáticas, concretamente nas variáveis String.

O que se considera como uma estrutura de dados estática, segundo o programa da disciplina, na Programação em Linguagem C, corresponde a dois conceitos que efetivamente partilham a mesma base que os define – Strings e Arrays.

Sendo um dos objetivos deste módulo introduzir o conceito de estruturas de dados estáticas como mecanismo de armazenamento de dados (DGFV, 2005), trata-se nesta fase de dar capacidades ao aluno para usar tudo o que aprendeu antes, numa perspetiva um pouco mais complexa no ponto de vista dos recursos computacionais disponíveis, não só para o armazenamento, mas também para a organização e manipulação de informação, quer pela utilização direta de funções da biblioteca da linguagem, quer por algoritmos a criar pelo aluno, para uma finalidade concreta.

Desta perspetiva, há conhecimentos pressupostos e necessários nas competências já adquiridas pelos alunos e que incluem os três módulos anteriores, embora do terceiro não seja totalmente dependente, ainda:

- Módulo 1- Introdução à programação e algoritmia. Nesta fase é essencial a capacidade de representação mental do ciclo de um programa, de identificação das variáveis simples e dos operadores principais;
- Módulo 2 - Mecanismos de controlo e execução. Para que seja possível a criação ou interpretação de programas sobre estruturas de dados é automaticamente implícito o domínio, ou a noção clara, sobre implementações de mecanismos e ciclos de execução e repetição condicional, incluindo combinações destes;
- Módulo 3 - Programação estruturada. Neste módulo iniciou-se a abordagem a uma forma de programar de nível estruturado, através da modularização do código de um programa e das respetivas variáveis. No caso da linguagem C, até aqui adotada, a implementação de subprogramas através de funções permite que os exercícios e atividades nesta Intervenção tirem partido da possibilidade de divisão de problemas em partes mais simples, fomentando a capacidade de abstração e permitindo o desenvolvimento de projetos em equipa ou de problemas com alguma complexidade, como é o caso da nossa atividade Ardumotic. É, no entanto, possível, embora não desejável, consoante as competências de cada aluno deste nível (caso existam muitas lacunas nestas competências), contornar a utilização destes conceitos através da programação monobloco.

Concretamente no Módulo 4, a ordem de temáticas, segundo o programa da disciplina, inicia com a aprendizagem de Strings como ponte para a aprendizagem dos Arrays. Apesar de algum constrangimento didático resultante dessa ordem de conceitos apresentado em mais detalhe na “Análise crítica do programa”, pretende-se atingir os seguintes objetivos de aprendizagem:

- Distinção entre uma variável simples e uma variável estruturada;
- Diferenciar índice e valor indexado num Array;
- Saber o que é uma String;
- Saber manipular uma String com os métodos mais utilizados;
- Saber aplicar os conhecimentos adquiridos em situações/finalidades concretas.

3.3. Análise crítica do programa da disciplina

Através desta Intervenção pedagógica, é colocado em prática uma pequena parte do programa oficial da disciplina PSI, o currículo que representa e as suas orientações para esta área. Cabe aqui uma breve análise dos seus aspetos críticos, positivos e negativos.

Quanto às finalidades expostas, o programa da disciplina PSI, segundo DGFV (2005), caracteriza-a como indispensável ao sucesso pessoal e profissional “nesta área”. Apenas no sentido de complementar, atualmente devemos ver a programação como essencial em todas as áreas STEM, pois não deve ser vista apenas como uma disciplina de especialistas informáticos, mas como uma ferramenta e um recurso para todos os cientistas e engenheiros, ou técnicos. Apesar de as orientações metodológicas focaram as suas finalidades práticas e criativas, devemos claramente assumir e fomentar a sua inter e transdisciplinaridade, tal como referido no Despacho n.º 6944-A/2018, em que se reconhece a necessidade de articular os documentos curriculares para o ensino básico e o ensino secundário, tanto numa abordagem vertical como horizontal. Se as competências que desta disciplina se espera são para a vida, neste caso referenciando mesmo ao nível dos conceitos de Programação, efetivamente podem ser aplicadas nas várias fases e projetos de cada pessoa em áreas diferentes em que se encontre profissionalmente.

Quanto à lecionação desta disciplina e ao seu sucesso (ou insucesso), sem querer alargar, pois é um aspeto de grande análise e referido a seguir, uma das causas apontadas tem sido a extensão do programa. Esta questão tem sido reconhecida pela DGE (segundo o Despacho n.º 6944-A/2018), considerando esta extensão inibidora de consolidação de aprendizagens, do desenvolvimento de competências de nível mais elevado, bem como um obstáculo à inclusão de alunos com necessidades específicas dificultando práticas de diferenciação pedagógica. Neste sentido, e resultante do decreto-lei 55/2018, definem-se algumas diretivas de redução de carga horária total dos cursos profissionais, delegando a sua adaptação à gestão de cada escola, como referido nos pontos anteriores, relativamente à carga horária total do Curso. Assumindo esta questão como complexa, pois temos por um lado as dificuldades pedagógicas e por outro as necessidades do mundo profissional e as competências esperadas do Perfil dos Alunos, deve apenas referir-se que quanto à extensão do programa, e a sua relação com o sucesso pedagógico, pouco ou nada mudou concretamente.

Quanto aos conteúdos específicos, e particularmente nesta temática Strings, observando a ordem de conceitos apresentada no Módulo 4 temos o seguinte:

1. String

- Definição;
- Declaração e manipulação;

2. Arrays

- Definição
- Declaração e manipulação
- Iniciação
- Pesquisa sequencial; inserção e remoção de elementos;

- Ordenação
- Array multidimensional

A Intervenção aqui descrita irá incidir na temática Strings, no entanto, e independentemente das circunstâncias, ter-se-á inevitavelmente que abordar o conceito que se sucede no programa, o de Arrays. Este fato contraria a constatação em muitos manuais e programas de disciplinas de programação na linguagem C, por exemplo o de Ritchie (1988) ou o de Damas (1999) entre outros, nos quais essa ordem é inversa.

Apesar de o programa da disciplina ter nas suas orientações de competências a desenvolver, o domínio da programação orientada aos objetos, e abrange 3 módulos específicos nesse paradigma, constata-se que nos módulos iniciais, incluindo o 4, existe uma orientação clara para ser usada a linguagem C, entre outras, não orientadas a este paradigma de programação. É importante ainda realçar que a apropriação ou não dos conceitos iniciais de programação tem relação direta com o desempenho do aluno no decorrer de todo o curso, já que os módulos mais avançados dependem fortemente desses conceitos. Neste sentido, apesar da possibilidade de enriquecimento da visão do conceito de Paradigma de Programação, podem ocorrer como consequência o desinteresse pelo conteúdo ministrado, ou até mesmo uma certa aversão à disciplina, dada a extensão e complexidade de abordagem no primeiro ano de um curso.

Quanto à estrutura modular deste curso, tem na prática prós e contras. Parte-se do princípio que permite maior flexibilidade e respeito pelos ritmos de aprendizagem dos alunos (Agência Nacional para a Qualificação e o Ensino [ANQE], 2019), disponibilizando também referenciais para processos de Reconhecimento, Validação e Certificação de Competências. No entanto, tendo em conta a inexistência de

precedências entre os módulos, surge a questão de até que ponto a dependência do domínio nos conceitos básicos aumenta nas atividades mais complexas dos módulos mais avançados como no caso de Projeto de Software.

A avaliação modular deste programa coloca estas questões, e como exemplo, visto que existe a possibilidade de um aluno ser aprovado no módulo 4, sem ter tido aprovação no módulo 2, dos mecanismos de controle e execução, podemos evidenciar duas possibilidades nesta situação:

- Uma na perspectiva de o aluno ter a possibilidade de iniciar um nível de aprendizagem que pressupõe o domínio de outro mais básico, que esteja sumativamente reprovado. Como se irá interpretar as lacunas dos conhecimentos em falta?
- E outra em que eventualmente o aluno seja capaz de dominar o módulo mais avançado que implicitamente inclui e envolve o domínio dos conceitos mais básicos, correspondentes ao Módulo anterior reprovado. O aluno irá ainda ter de repetir o dito Módulo reprovado?

Sendo que esta modularidade não tem precedências, o início de cada módulo torna-se em alguns casos paradoxal, no entanto com possibilidade ser positivo, dependendo da abordagem. Se for feito um diagnóstico aos conhecimentos de cada aluno no início de cada módulo, é possível transformar o que era uma condenação na inspiração de um reinício com novos desafios. Cabe ao Professor essa transformação. E neste sentido, sendo que no programa da disciplina consta a referência à avaliação diagnóstica apenas no início do ano letivo, devia considerar-se uma maior ênfase na avaliação diagnóstica contínua ao longo do ano letivo (que na verdade seria formativa),

mais propriamente no início de cada módulo de modo a adaptar as atividades e temáticas às necessidades e potencial de cada aluno.

O programa da disciplina indica que se deve privilegiar a participação dos alunos em projetos e na resolução de problemas e de exercícios que simulem a realidade (DGFV, 2005), no entanto carece de orientações mais concretas ao nível de recursos, metodologias e exemplos de atividades para “aplicações práticas e contextualizadas das competências, a experimentação, a pesquisa e a resolução de problemas”. Para isso é necessário o estímulo para as atividades extras curriculares, e facilitar o acesso a recursos, deslimitando-os dos períodos, locais e finalidades letivos, promovendo também a transdisciplinaridade.

Tendo em conta esta necessária flexibilidade e considerando as áreas STEM (Science Technology Engineering and Mathematics), o Pensamento Computacional começa a ser automaticamente introduzido, embora o seu conceito seja ainda considerado por alguns como “muito vago” por “raramente ser possível diferenciá-lo de outras formas de pensamento” (Tedre, 2016). No entanto, se efetivamente o ensino de Programação é uma forte estratégia para desenvolver as competências dos alunos no pensamento computacional (Piedade et al., 2019, p. 1) , a disciplina de PSI aqui focada, como primeiro contacto formal destes alunos com a Programação, deve considerar-se não só em toda a problemática de ensino, mas também no seu potencial de aplicação de inter e transdisciplinaridade.

3.4. Problemática e estado atual no Ensino da Programação

Nas últimas décadas, as novas gerações, conhecidas como “nativos digitais”, segundo Prensky (2001), deparam-se nesta mudança social e cultural, também com

um conflito de metodologias e paradigmas de ensino, onde os métodos e recursos tradicionais vão sendo “empurrados” pelos digitais.

Propriamente na área da programação, o ensino regular desta disciplina reconhece-a como das mais problemáticas. A investigação educacional tem vindo a referir várias causas para o insucesso e dificuldades dos alunos, mas antes de focar essas causas estudadas, convém referir que Programação em si constitui-se em várias dimensões que a formam num todo como disciplina. Segundo Piedade et al. (2019) a aprendizagem de programação implica mobilização de conhecimentos e competência em computação, algoritmia e pensamento lógico, identificação de problemas, design e codificação de soluções, na compreensão da sintaxe e semântica de uma linguagem e o domínio ou conhecimento de paradigmas de programação. Sendo esta visão holística muito importante para a perceção deste cenário, vejamos mais algumas perspetivas.

Partindo da constatação de que existem várias metodologias “alternativas” para o ensino da Programação, e uma grande variedade de escolha de recursos, que vai desde as interfaces visuais e por blocos, aos robots pedagógicos e plataformas de prototipagem como o Arduíno Drones, no entanto reside a questão, “Qual a finalidade e grau de exigência e abstração esperado dos alunos nas atividades? E de que forma adequá-las ao ritmo e perfil de cada um?”. Com certeza a lista de finalidades do programa e as competências esperadas do currículo servem apenas de referência para responder a esta questão e muitas vezes são incompatíveis com o nível de maturidade e conhecimento real dos alunos, desviando o docente de uma estratégia efetivamente adequada.

Das dificuldades constatadas, estudos como o de Santos et al. (2005) apontam para uma discrepância entre complexidade e extensão de conteúdos da programação e

a sua aplicabilidade real, acentuada pelo fato de o ensino tradicional favorecer a aprendizagem abstrata, ou seja geralmente desligada do contacto com o mundo físico envolvente.

Segundo Price et al. (2002), quanto à motivação, a complexidade das linguagens de uso profissional pode ser um fator negativo à partida devido à exponencialidade da curva de aprendizagem e esforço exigido. E quando se fala de motivação na aprendizagem é necessário encontrar e aplicar a metodologia mais adequada e os respetivos recursos e estratégias para cada contexto. Neste objetivo, o professor depara-se com um dilema comum, por um lado as linguagens de programação disponíveis vão de um leque que contém as de nível profissional (Java, C++,etc.) às mais pedagógicas e simplistas com interface visual ou por blocos. Por outro, a tendência do professor é escolher aquela que domina e considera que ele próprio terá melhor desempenho a lecionar (Piedade et al, 2019).

Do estudo bastante abrangente de Luxton-Reilly et al. (2018), e considerando uma fase inicial da aprendizagem de Programação, como é aqui o caso, existe também a possibilidade problemática do elevado grau de expectativa que pode resultar numa abordagem de elevada dificuldade na forma de exposição de conceitos por parte do professor, mesmo que estes até sejam simples, devido a falta de literacia de código dos alunos.

Segundo este autor, a literacia de código, propriamente as capacidades de leitura de código estão intimamente relacionadas com as capacidades de resolução de problemas e a escrita de código. Considera-se também que para o Professor é muito importante a observação do processo de “code-tracing” de cada aluno, pois isso revela

o mecanismo cognitivo que utiliza na interpretação de código, mais propriamente o que Luxton-Reilly chama de “make sense of code”.

Esta literacia que não é mais que o domínio da escrita e leitura de uma linguagem de programação propriamente dita, tal como no seu sentido mais comum, exige hábito e treino. Se for encarada de uma forma pura e crua pode transformar-se em algo demasiado abstrato, e para que possa fazer sentido para os alunos deve haver uma aplicação significativa da mesma em contextos reais.

Em Portugal, em termos de políticas na tentativa de mitigar o que à Informática e à Programação diz respeito, foi lançado em 2015 pela DGE, um projeto piloto para promover a Iniciação de Programação no 1º Ciclo de Ensino Básico, nas ofertas Complementares e nas atividades de enriquecimento curricular (DGE, 2020). Nesta tendência global, e em forma de resultados do ICILS 2018, apresentados no Websumit 2019, a literacia digital dos alunos a frequentar o 8.º ano de escolaridade, ocupa o 7º lugar da escala do CIL entre 14 países e regiões participantes, situando-nos assim a meio da tabela.

Como avaliação curricular em 2018 pela DGE, do nosso ensino básico e secundário, segundo o despacho 6944-A/2018 de 19 de Julho, reconhece-se também o problema da extensão dos programas e metas curriculares. Facto este identificado sobretudo pelos professores, revelava-se inibidor de consolidação de aprendizagens, do desenvolvimento de competências de nível mais elevado. Acrescenta-se ainda que os documentos curriculares para o ensino básico e o ensino secundário, aplicados ao longo das últimas três décadas, careciam de articulação entre si, tanto numa abordagem vertical como horizontal de interdisciplinaridade.

Efetivamente, nas últimas décadas a Programação tem sido automaticamente inserida nas áreas da ciência, tecnologia, engenharia e matemática. E a razão óbvia é que, à medida que os computadores se tornam uma ferramenta indispensável para os cientistas e engenheiros, eles vão precisar de entender e desenvolver programas computacionais específicos para poderem continuar o desenvolvimento na sua área (Rubio et al., 2013).

Uma das formas de ensino nas áreas STEM, e focando a sua relação com a Programação, é através de resolução de problemas (concretamente pela metodologia Problem Based Learning), estimulando o pensamento crítico e a criatividade. Nesta perspetiva, vários estudos indicam que as capacidades criativas dos alunos estão ainda num nível moderado ou fraco (Omar et al., 2017), (Usta & Akkanat, 2015). Talvez uma das causas para este nível de criatividade aquém do desejado seja, segundo Hoffer (2012), o facto de o currículo geralmente segmentar demasiado as áreas científicas impedindo que os alunos constatem e observem como é que na realidade essas áreas se interligam e funcionam em conjunto.

4. Intervenção Pedagógica

Neste capítulo são apresentados os seguintes pontos que permitiram a materialização das bases para a concretização desta Intervenção:

- Definição dos conteúdos de lecionação – Os conceitos científicos e a respetiva temática a abordar;
- Descrição dos principais objetivos desta PES, abrangendo os conteúdos abordados na sua relação com as competências esperadas nos alunos, e tendo em conta toda a conjuntura até aqui descrita.
- A metodologia e estratégias que estruturam a preparação de uma experiência positiva de aprendizagem;
- Os planos concretos para cada aula que permitem ao professor uma orientação segura e eficaz;
- E finalmente a descrição da sua operacionalização.

4.1. Fundamentação e enquadramento conceptual

Até à fase de realização da PES, nesta disciplina, os alunos já tiveram contacto implícito com a utilização Strings, por exemplo através das funções de input/output *printf*. e *scanf*, nas quais são enviadas ou recebidas Strings de caracteres para o ecrã ou do teclado respetivamente. No entanto a sua perceção sobre este tipo de variável não passa da observação da sua utilização implícita na finalidade pretendida.

Subdividindo interpretação do termo String, por exemplo uma sequência de caracteres entre aspas, como “*Hello world\n*”, é chamada String de caracteres (no sentido de “sequência” de elementos do mesmo tipo char) ou Constante String (Ritchie, 1988), tal como o valor ‘A’ de uma variável *char* é chamada uma Constante

Char. Ou seja, quando os alunos chamam a função *Printf*("hello, world\n") estão efetivamente a colocar no argumento da função *Printf* uma String de caracteres (ou Constante String).

Na programação de computadores o termo String refere-se ao um tipo de variável, que representa um array que contém uma String de caracteres, e chamamos então a esse array Variável String.

Apesar de o conceito e definição semântica de String ser o mesmo nas diversas linguagens de programação, existem diferenças importantes no que respeita à forma da sua criação e manipulação. Na linguagem C, a String corresponde simplesmente a um vetor que contém uma sequência de caracteres, mas na linguagem C++ as Strings são criadas e utilizadas a um nível superior de abstração de Classe/Objeto, mais propriamente no paradigma da programação orientada aos objetos.

Perante esta definição, e visando a lecionação da disciplina PSI com todo o planeamento preparado, organizam-se os principais conceitos científicos centrados nas variáveis String da seguinte forma:

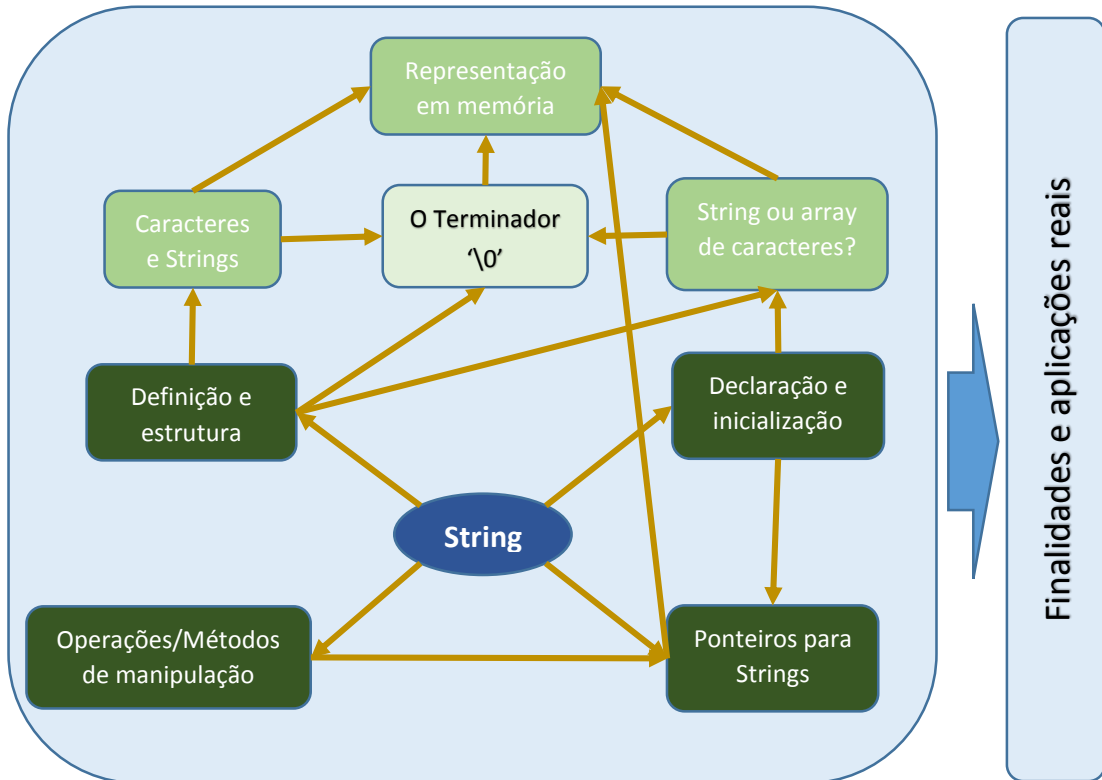


Figura 4 - Diagrama de conceitos global da temática Strings

Neste diagrama considera-se que o tipo de variável String como conceito central, deriva em 4 conceitos básicos indissociáveis:

- A “definição e estrutura”, que implica automaticamente a distinção entre caracteres (variáveis char) e Strings, assim como entre Strings e arrays de caracteres. Em ambos os casos esta distinção é baseada no elemento/conceito Terminador ‘\0’;
- A “declaração e inicialização de Strings”, que implicitamente envolve também a distinção entre String e array de caracteres, tendo em conta a dimensão e os

carateres inicializados (ou não), como se pode ver no ponto seguinte relativo a este conceito;

- Operações e métodos de manipulação. Corresponde às funções predefinidas da biblioteca do Ambiente de Desenvolvimento da linguagem C para executar operações de manipulação de Strings, ou às funções manualmente construídas para essa finalidade.
- Ponteiros para Strings. Corresponde a uma das formas de criação e representação de Strings, através de ponteiros, cujo endereço de memória para o qual aponta pode mudar dinamicamente.

A representação de Strings na memória do computador é o conceito que falta referir, é muito importante e complementar para a perceção da estrutura da String, assim como para a sua distinção das variáveis *char* e dos arrays de carateres. Mais ainda, a visualização do representação das Strings em memória permite também discernir mais facilmente o conceito de ponteiro para String, muitas vezes de difícil assimilação pelos alunos.

4.1.1. Definição de String

No âmbito do ensino de Programação, traduzido no nosso programa curricular sequencial modular, para se definir o tipo de dados String existe um conceito preliminar necessário de abordar, ou recapitular, o conceito de array, na sua simples definição como vetor de elementos do tipo de dados simples, como *int* ou *char* por exemplo.

Exemplo:

Char vetordecarateres[10]={‘a’, ‘e’, ‘i’, ‘o’, ‘u’};

Segundo Damas (1999), e como veremos, uma variável String corresponde sempre a um vetor de caracteres, mas um vetor de caracteres pode não corresponder necessariamente a uma String. Essa diferença que os distingue acontece no momento da Declaração e Inicialização como veremos no ponto seguinte.

Em C as Strings não são um tipo básico de dados e a única forma de representar uma sequência de caracteres é recorrendo a um vetor de caracteres (Damas, 1999). Então como distinguimos um vetor de caracteres daquilo a que pretendemos chamar String, mas que na verdade é também um vetor de caracteres?

Imaginemos a Constante String “Ana” que queremos armazenar num array de 5 elementos. À partida aquilo que precisávamos é de um array com dimensão suficiente para este conjunto de caracteres recebido. Mas também é necessário ser possível identificar esta sequência de caracteres no array, do primeiro ao último sequencialmente.

Para identificar a nossa constante String no vetor, assumindo como regra geral que começa sempre no elemento zero do vetor, e que a dimensão do vetor é superior ao número de caracteres, é um terminador que permita ao compilador localizar o final da String no vetor. Escolheu-se pelas razões naturais que esse terminador seria elemento ‘\0’ da tabela ASCII mais propriamente o seu elemento zero também designado NULL. Para além de ser colocado o terminador ‘\0’ a seguir ao último carácter, é também colocado esse mesmo valor nos restantes elementos não utilizados, nesse caso com o significado de “valor não definido” ou NULL.

char Nome[5] = “Ana”;

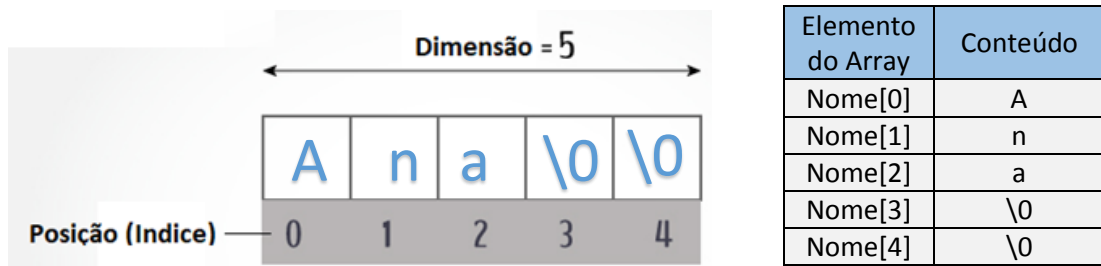


Figura 5 - Representação da estrutura de uma String

Existe muitas vezes a tendência de resumir a definição de String a uma sequência de caracteres terminada em ‘\0’, no entanto e principalmente no ensino de programação é necessário alguma prudência e coerência na apresentação dos seus vários aspetos para evitar erros comuns dos alunos. Alguns desses aspetos são por exemplo não serem possíveis atribuições diretas de valor a variáveis String após a sua inicialização, ou atribuição de conteúdo de uma String a outra, como por exemplo:

Declaração:

```
char nome[20];
```

Declaração e Inicialização:

```
char segundonome[20]="Pires";
```

Expressões erradas após a declaração ou declaração com inicialização:

```
nome ="joana";
```

```
segundonome=nome;
```

4.1.2. Caracteres e Strings

Na linguagem C, e por este ser um aspeto que causa inicialmente alguma confusão nos alunos, é bom abordar a diferença entre uma variável char e uma String com uma letra apenas, apesar de o conteúdo ser o mesmo. Enquanto o char ocupa

apenas um byte, correspondente ao seu valor na tabela ASCII, a String ocupa 2 bytes, respetivamente um para o carater, tal como o anterior, e outro para o elemento terminador de String '\0', como se vê na próxima ilustração.

Sendo esta a forma mais básica de exemplificar ambos os tipos de dados, char e Strings respetivamente, e mesmo sendo igual o conteúdo em termos de caracteres, são distintos por exemplo na inicialização, em que as Strings recebem o seu valor entre aspas, e os caracteres recebem o seu valor entre plicas. Mais ainda, a String mesmo que contenha apenas um carater usa a mesma declaração dos vetores, ou seja, com os parênteses retos para a dimensão.

Char letra1='a';

Char string1[]="a";

A outra diferença, e importante de explicar é a representação de Strings e caracteres na sua estrutura em memória:

Tabela 5 - Representação em memória de uma String e uma variável char.

```
char letra1='A'
char string1[]="A";
```

Endereço	Conteúdo	Variável
...
2999	A	letra1
3000		
3001	A	string1
3002	\0	
3003		
3004		
3005		

Pode verificar-se que o vetor `string1[]` tem dimensão 2, que foi definida neste caso automaticamente, calculando a dimensão da String com um carater mais um para o terminador `'\0'`.

4.1.3. Declaração e Inicialização

Partindo da constatação de que o que distingue a String de um simples array de elementos char é o fato de a respetiva sequência de caracteres nele guardada ser finalizada com o elemento `'\0'` da tabela ASCII, vamos analisar as possíveis situações em que na prática ambos se podem confundir ou distinguir.

Na linguagem C declarar uma String é efetivamente declarar um vetor de caracteres, e se não for inicializado, não existe distinção, independentemente da sua finalidade, como por exemplo:

```
Char nome[10]; //supostamente para guardar uma String de caracteres
```

Ou:

```
Char vetordecarateres[10]; // supostamente para guardar um conjunto de caracteres
```

Em ambos os casos, como não houve inicialização, todos os elementos são colocados a NULL (`'\0'`).

A inicialização de Strings consiste em atribuir conteúdo ao vetor de caracteres no momento da sua declaração, mais concretamente uma sequência de caracteres terminada em `'\0'`. Deve referir-se que esta inicialização, através da atribuição direta com o operador `"="` apenas é possível na sua declaração.

Existem então dois parâmetros que são definidos nesta fase, o conteúdo e a dimensão do vetor que contém a String. Estes dois parâmetros devem ser bem definidos consoante a finalidade a que se destinam, caso contrário poder-nos-emos deparar com ambiguidades e erros inesperados de execução.

Focando esses dois parâmetros, vejamos alguns exemplos que nos permitem constatar as diferenças e semelhanças entre Strings e vetores de caracteres, assim como algumas dessas ambiguidades possíveis na sua inicialização:

- **O conteúdo:**

A forma de declarar o conteúdo define se estamos a criar uma String ou um simples array de caracteres.

Exemplo 1: Criação e inicialização de uma String num vetor de 20 elementos:

```
char nome[20] = "Andre";
```

Neste caso é automaticamente colocado o elemento ‘\0’, especificamente como terminador de String, a seguir ao último carater assim como nos seguintes não utilizados do vetor.

Exemplo 2: Criação e inicialização de um vetor de caracteres de 20 elementos:

```
char nome[20] = {'A', 'n', 'd', 'r', 'e'};
```

Neste caso foram inicializados os primeiros 5 caracteres e todos os restantes definidos a Null (‘\0’) por não serem utilizados.

Ou seja, nestes dois primeiros exemplos não existe diferença entre uma String e um vetor de caracteres pois o conteúdo é exatamente o mesmo e apesar de a inicialização ter usado um formato diferente.

- **A dimensão:**

Pode ser definida explicitamente como nos exemplos anteriores ou automaticamente. Se a dimensão for definida explicitamente, então não é obrigatório inicializar o conteúdo da String, e nesse caso todos os seus elementos ficam com o valor NULL (`'\0'`) por defeito.

Mas se a dimensão não for explicitamente definida então o seu conteúdo terá de ser inicializado, caso contrário não será possível compilar o código. E neste caso a dimensão será atribuída consoante essa inicialização. Vejamos os seguintes exemplos:

Exemplo 3: Inicialização de uma String sem definir explicitamente a dimensão:

```
char nome[]="Andre";
```

Neste exemplo o compilador calcula o número de caracteres existentes na String para criar um vetor com a dimensão do número de caracteres mais o terminador `'\0'`, resultando assim num vetor de dimensão 6;

Exemplo 4:

```
char vetordecarateres []={'A','n','d','r','e'};
```

Neste caso estaríamos a criar explicitamente um vetor de caracteres de dimensão 5, e inicializado com os 5 caracteres individualmente definidos. Podemos verificar essa dimensão através da função *sizeof(vetordecarateres)*.

Se compararmos com o exemplo anterior constatamos efetivamente a diferença na interpretação pelo compilador ao distinguir se está a criar uma String ou um array de caracteres. Neste caso foi criado um array com dimensão específica para os caracteres inicializados e apenas esses, ou seja sem terminador `'\0'` a seguir ao último como aconteceu no exemplo anterior.

Exemplo 5:

```
char nome[5]="Andre";
```

Este exemplo serve para visualizar o que aconteceria se o número de caracteres a inicializar na String for igual ou superior à dimensão definida explicitamente no vetor que a contém. Neste caso não irá haver “espaço” para colocar o elemento terminador ‘\0’. Ou seja, na prática teríamos o mesmo que no exemplo anterior em que foi criado explicitamente um vetor de caracteres, resultando assim numa ambiguidade. Temos neste caso uma String ou um array de caracteres? O resultado é um array.

Na prática, na linguagem C é possível executar programas a usar funções de Strings sobre vetores de caracteres apesar das possíveis ambiguidades, quer tenham sido inicializadas explicitamente como String, ou como array de caracteres, com ou sem o elemento terminador ‘\0’. Isto porque para o compilador uma String é efetivamente um array, embora as funções da biblioteca para manipulação de Strings considerem a característica adicional esperada que as caracteriza e que é o terminador ‘\0’, mas na falta desse terminador é considerado o último elemento do vetor como final da String. Se a estrutura das Strings no programa perderem a coerência, erros inesperados poderão acontecer nos resultados consoante os algoritmos utilizados e a forma como trabalham com a terminação e dimensão das Strings.

Resumindo, tendo em conta esta partilha de estrutura entre vetor de caracteres e String, e assumindo que se considera uma String apenas quando terminada em ‘\0’, afirma-se que, de acordo com Damas (1999), um vetor de caracteres pode não conter uma String.

4.1.4. Strings e ponteiros para Strings

Existe ainda uma outra forma de criação de Strings, e de aceder ao seu conteúdo, através de ponteiros do tipo `char`. O conceito de ponteiro é incontornável nesta temática, pois ele está implícito por exemplo em algumas funções que retornam Strings pelo seu endereço e não pelo conteúdo, o que tem a vantagem de tornar a execução dos programas mais rápida e eficiente.

Vejamos o seguinte exemplo de um ponteiro para String com respetiva inicialização:

```
char* nome = "Andre";
```

Neste exemplo existem dois elementos distintos, um endereço de memória, associado à variável *nome*, e uma String armazenada nesse endereço de memória.

Deve salientar-se que *nome* não é uma String, mas existe uma String no endereço para o qual o ponteiro *nome* aponta.

A sua representação em memória é a seguinte.

Tabela 6 - Representação em memória de um ponteiro para String

Endereço	Conteúdo	Variável
...
2999	3001	nome
3000		
3001	A	
3002	n	
3003	d	
3004	r	
3005	e	
3006	\0	

Sendo que o valor de um ponteiro para String é apenas um endereço, tem como vantagem poder “apontar” para o endereço de outra String existente no programa, atribuindo-lhe simplesmente o endereço dessa String.

Vejamos o exemplo com a seguinte sequência de 1) a 3):

Tabela 7 - Representação em memória de uma String, um ponteiro para String e uma atribuição do endereço da String ao ponteiro.

1)			2)			3)		
<code>char Str1[]="ABC";</code>			<code>char Str1[]="ABC";</code> <code>char* str2;</code>			<code>char Str1[]="ABC";</code> <code>char* str2;</code> <code>str2=str1;</code>		
Endereço	Conteúdo	Variável	Endereço	Conteúdo	Variável	Endereço	Conteúdo	Variável
...
2999	A	str1	2999	A	str1	2999	A	str1
3000	B		3000	B		3000	B	
3001	C		3001	C		3001	C	
3002			3002			3002		
3003			3003	NULL	str2	3003	2999	str2
3004			3004			3004		
3005			3005			3005		

As seguintes duas instruções têm exatamente o mesmo output, ou seja, em termos de acesso ao conteúdo, quer a String quer o ponteiro para String podem ser interpretados da mesma forma quando usamos o *printf*.

```
printf("%s",str1); // output: ABC
```

```
printf("%s",str2); //output: ABC
```

Observar que, no caso do ponteiro `*str2`, para acedermos ao conteúdo da localização apontada chamamos a variável `str2` sem o asterisco;

Um das vantagens dos ponteiros é que podemos alterar o conteúdo simplesmente alterando o endereço para onde aponta. Ou no caso das funções, quando se espera uma String no retorno, passar apenas o endereço dessa String, tornando a execução mais rápida:

Tabela 8 - Representação em memória de uma String e um ponteiro para String

Endereço	Conteúdo	Variável
...
2999	A	Str1
3000	B	
3001	C	
3002		
3003	3006	Str2
3004		
3005		
3006	D	Str3
3007	E	
3008	F	

```
Char str3[]="DEF";  
Strcpy(str2,str3);
```

4.2. Objetivos da leção e aprendizagem na PES

Segundo Ferraz e Belhot (2010), na educação, decidir e definir os objetivos de aprendizagem significa estruturar, de forma consciente, o processo educacional de modo a oportunizar mudanças de pensamentos, ações e condutas. Esta definição, no aplicável a um ponto de partida como o desta fase de aprendizagem de Programação, é provavelmente uma das mais adequadas, pois o termo “oportunizar”, no seu significado de “abertura bilateral”, engloba e prevê quer as potencialidades cognitivas (tendo em conta o seu estágio de desenvolvimento subjetivo) de cada aluno mas também as possibilidades que os cenários/atividades possíveis atualmente podem proporcionar em conhecimentos significativos numa perspectiva construtivista.

Concretamente esta definição de objetivos, que deve partir da consideração da aquisição de conhecimentos e de competências esperadas neste perfil profissional específico, determina os pontos orientadores que se seguem, relacionados com a escolha de metodologias, estratégias, recursos, atividades e instrumentos de avaliação a ser adotados por um determinado período de tempo.

No que ao aluno se refere, o termo objetivo aqui descrito vai além do que se espera que os alunos aprendam, abrangendo também e essencialmente o que se espera eles sejam capazes de realizar com esse conhecimento, ou seja as competências adquiridas. Para isso, e visando as finalidades curriculares estabelecidas e os objetivos apresentados no ponto anterior de caracterização do módulo lecionado, como ponto de partida pretende-se que;

- A planificação das aulas no seu todo seja integradora e o mais abrangente possível dos conhecimentos já adquiridos;

- Os exercícios dedicados à assimilação e domínio de conceitos da primeira parte da PES, complementados com a atividade prática resultem numa aprendizagem sólida e significativa proporcionalmente às competências, ao proporcionar uma experiência de aplicação e utilidade real, no sentido de os alunos se aperceberem que poderiam aplica-la no seu dia-a-dia, observando limitações, constrangimentos e potencialidade daí resultantes.

Em termos letivos e considerando as capacidades cognitivas esperadas nos alunos, pelo facto de os conceitos desta disciplina exigirem uma relativamente elevada capacidade de abstração, e um tipo de raciocínio relativamente diferente daquele a que estão habituados no dia-a-dia, deve elaborar-se os exercícios, quer da primeira quer da segunda parte da PES, com certos limites de exigência e planeamento de fases distintas de evolução da aprendizagem, pois essa capacidade de abstração dos alunos pode ainda estar aquém do esperado pelo docente.

Concretamente as competências principais esperadas desta aprendizagem, em conformidade com o proposto no programa da disciplina (DGFV, 2005) são:

- Saber escolher e adequar as soluções tecnológicas aos problemas a resolver;
- Conceber algoritmos através da divisão dos problemas em componentes;
- Estimular a reflexão, a observação e autonomia;
- Saber escolher e/ou identificar a arquitetura da solução mais adequada aos problemas;
- Estimular o raciocínio lógico;

4.3. Metodologias de ensino

Considerando como ponto de partida o programa estabelecido para esta disciplina, devem privilegiar-se metodologias através de atividades que incidam sobre a aplicação prática e contextualizada dos conteúdos, a experimentação, a pesquisa e a resolução de problemas (DGFV, 2005).

Associando a conjuntura atual do ensino de programação e este contexto específico à importância das áreas STEM no currículo, para que o seu ensino seja eficiente, considerando como um exemplo de referência a experiência de Jamal et al. (2020), as competências cognitivas mais importantes que os alunos precisam desenvolver são o pensamento crítico e o pensamento criativo. Mas, um ingrediente essencial e necessário à partida nesse desenvolvimento e para o sucesso da aprendizagem, segundo Luxton-Reilly et al. (2018) é a auto motivação e possibilidade de se apropriar do contexto/cenário das atividades propostas.

Citando o Prof. Alessandro Bogliolo (Bogliolo, 2020), consultor na criação de robôs programáveis para crianças com mais de 8 anos, embora no contexto paralelo da robótica educativa mas também aplicável na metodologia PBL e às atividades com microcontroladores, “A construção e a programação de um robô são atos criativos que exaltam a fantasia e engenho.”.

Assumindo esta metodologia como orientação das principais estratégias consideram-se duas diretivas principais:

- A resolução de problemas é um processo que envolve manipulação de/ou operações com conhecimento previamente adquirido;
- A evolução da capacidade de abstração de um conteúdo a partir do estímulo do desenvolvimento cognitivo linear, é partir de conceitos mais simples para os

mais elaborados. (Ferraz & Belhot, 2010). Ou seja, pretende-se proporcionar uma aprendizagem cuja aplicação real tenha uma complexidade de evolução gradual em termos de quantidade de código e variáveis necessárias, e se possível que as várias questões/desafios formem um conjunto modular, ou seja, cada uma conduza à resolução de uma parte importante do todo, que no final será possível observar de forma integrada.

Quanto ao tipo de problemas a apresentar, variam consoante a estrutura na qual os mesmos são apresentados (Kirkley, 2003). Seguindo esta visão e segundo Newell & Simon (1972) podemos assumir como referência três níveis de problemas propostos, para os quais se apresenta um exemplo relacionado com esta temática na Programação:

- Bem estruturados - Consideram-se os problemas que têm apenas uma solução ou são normalmente resolvidos da mesma forma e com recursos/técnicas facilmente dedutíveis. Exemplo possível: “*Complete as palavras que faltam no script, com a respetiva função de manipulação de Strings*”;
- Moderadamente estruturados - São o tipo de problemas para os quais pode existir mais que uma solução de um leque de possibilidades, e onde as estratégias e recursos necessários são razoavelmente acessíveis e pré-disponibilizados. Exemplo: “Identifique uma função equivalente ao *fgets* para leitura de Strings do teclado, para obter o mesmo resultado de execução, e se for o caso apresente as diferenças observadas”;
- Pouco estruturados. São o tipo de problemas para os quais podem existir várias formas de abordagem e de resolução, não necessariamente todas previstas. A orientação neste tipo de problemas também não é totalmente preparada, ficando a cargo do aluno delinear estratégias. Como exemplo, relacionando

com a atividade prática Ardumotic: “Crie uma função para executar um scroll letf da String a apresentar no LCD quando essa String é maior que a sua capacidade fixa em número de carateres”.

O nível em que se encontra o problema apresentado e a sua adequabilidade ao estágio de conhecimentos e competências dos alunos nessa fase irá determinar a forma como vai decorrer a aprendizagem e concretização de objetivos.

De um ponto vista comum a todos os níveis de estruturação, e isolando o fator “qualidade da elaboração dos problemas” que depende das competências do professor, parte-se do princípio que esta continuidade de níveis tem a ver com o contexto didático e com os objetivos definidos, como referido no ponto anterior. Desta forma, geralmente os problemas bem estruturados estão associados a um baixo nível de dificuldade, e os pouco estruturados estão associados a um nível de dificuldade mais elevado. Neste ultimo caso, presume-se que os alunos já adquiriram competências de autonomia e domínio dos conceitos básicos necessários para definir as próprias estratégias e usar a criatividade.

Nas estratégias concretas apresentadas nesta PES escolheu-se o nível de problemas Moderadamente Estruturado.

4.4. Planificação

Planificar é definir as linhas orientadoras da execução prática das aulas a lecionar, tendo em conta os principais objetivos definidos, seguindo orientações das várias dimensões até aqui descritas, desde o enquadramento global aos recursos disponíveis, no intuito de transformar esta experiência em aprendizagens significativas. Esta planificação reflete os objetivos globais da PES, sendo que os

objetivos mais específicos agora apresentados tem como critérios base a clareza de apresentação relativamente aos conteúdos/conceitos e competências esperadas, a coerência e adaptação da sequência das atividades propostas ao real enquadramento cognitivo e socio-afetivo dos alunos da turma.

Para os quatro dias de aulas realizadas correspondentes a 14 blocos de 45 min os respetivos documentos com a planificação detalhada podem ser consultados nos anexos E1, E2, E3 e E4.

4.4.1. Sobre os principais recursos utilizados

Para a escolha dos dois recursos educativos principais, a IDE DEV C++ e a plataforma Arduíno, para garantir a sua complementaridade, foram ponderadas as diferenças e semelhanças no ambiente de desenvolvimento e a sintaxe da linguagem, assim como a versão do seu compilador, de maneira que a chamada das funções sobre Strings seja o mais idêntica possível.

O DEV C++ utiliza o compilador GCC do GNU GCC, suportando as linguagens de programação C e C++, e possui toda a biblioteca ANSI C. O Arduino assenta também numa versão do compilador GCC, mais propriamente o AVR-GCC criado para os microcontroladores Atmel (AVR-Libc, 2020) disponibilizando o pacote de livrarias AVR-libc 2.0.0 - Standard C. Ou seja, toda a sintaxe prevista mais utilizada no DEV C++ é também possível de utilizar da mesma forma no Arduino.

Um aspeto importante de referir e que distingue estes recursos, é que o Arduíno é um sistema embebido. No domínio da engenharia, este tipo de sistema não tem por finalidade a computação genérica, no entanto é controlado por um sistema computacional. Quanto à sua aplicabilidade, a possibilidade de ligar o mundo físico ao

mundo digital com este tipo de controladores computacionais tem atualmente uma grande dimensão, que vai desde aplicações domésticas e de automação até à indústria automotiva (El-Abd , 2017).

Sendo a plataforma Arduíno das mais comprovadas no seu êxito, embora isso seja relativo à finalidade proposta, teve-se também em consideração os indicadores de perfil dominante dos alunos desta turma no que se refere à dinâmica mais produtiva em sala de aula, que é em trabalho de grupo e com aplicações interativas.

Sendo que esta plataforma permite contextos quase ilimitados de utilização e desenvolvimento de competências de programação, identificaram-se duas formas de tirar partido dela. A primeira através da utilização de um display para visualizar informação recolhida de um módulo de captação de temperatura e humidade (DHT11), e a segunda através do próprio Monitor-Série da sua IDE, que permite o envio e receção de informação, via Serial-Com entre o Arduíno e o computador ao qual está ligado, através de Strings de caracteres.

Os dispositivos eletrónicos utilizados são modulares e de ligação simples, e o seu interface com a programação é baseado em funções de uma biblioteca fornecida que permitem ao utilizador preocupar-se apenas com a programação genérica.

Dada a extensa aplicabilidade do Arduíno, e o fato de ser uma plataforma de prototipagem, deve ser utilizado metodicamente e em pequenas atividades práticas demonstrativas e experimentais cuja complexidade aumente gradualmente, de forma a estimular os alunos a pesquisar e expandir os seus conhecimentos com confiança e autonomia, pois são possíveis muitos erros e problemas de montagens, ligações elétricas, etc.

Um dos benefícios de programar com o Arduino é o facto de ser OpenSource permitindo que todos os alunos tenham acesso gratuito ao seu software assim como a muitos programas disponibilizados pela comunidade de desenvolvimento.

Segundo Rubio et al. (2013), no seu estudo de utilização do Arduino para o ensino de Programação em cursos de ciência e engenharia, é possível obter melhores resultados de aprendizagem e satisfação dos alunos em relação aos métodos tradicionais com a utilização do Arduino. Este estudo usou uma estratégia parecida com a aqui presente, no sentido de usar o Arduino como complemento à abordagem tradicional. No entanto os resultados do estudo, quanto à satisfação e aprendizagem advêm da análise comparativa dos resultados entre duas turmas, de 2 anos letivos consecutivos respetivamente, numa praticou-se a aprendizagem “tradicional” de programação e na outra utilizou-se o Arduino. Como exemplo, das atividades das aulas com Arduino, nas sessões de laboratório deste estudo ministrava-se os conceitos de arrays utilizando speakers, ou seja a melodia final corresponde ao conjunto de “notas musicais”, cada uma num elemento do array, possibilitando também concatenações e outras operações sobre esses arrays.

4.4.2. Estratégias

A realização dos planos de aulas envolvem na prática métodos e recursos diversificados. Para aplicar a metodologia PBL nesta temática definiram-se as seguintes orientações estratégicas gerais, para as quais se considera como referência a experiência de Beghetto (2009):

- Inspirar e motivar os alunos através de situações/desafios adequados ao seu perfil com aplicações reais e contextualizadas de conhecimento;

- Promover o envolvimento dos alunos nas atividades da aula, pela Auto motivação e apropriação dos conhecimentos:
 - Não limitar as novas ideias apenas ao que consta nas diretivas do currículo ou do planeamento do professor;
 - Dar valor e consideração a todos os tipos de descobertas feitas pelos alunos;
 - Ter como princípio a possibilidade de todos os alunos serem criativos independentemente da sua classificação sumativa.
- Fazer os alunos pensar e refletir, apercebendo-se de obstáculos, dificuldades ou ambiguidades, e sentir a necessidade de os ultrapassar, mas com limites controlados dentro do possível:
 - Independentemente das finalidades, ajudar os alunos a entender antes de mais, os objetivos, a definir e racionalizar o problema subdividindo-o em partes;
 - Promover a criatividade e a determinação na resolução de problemas com os recursos e métodos adequados.

Resultando destes princípios orientadores dividiu-se as aulas disponíveis em duas partes, a primeira utilizando como recurso principal o ambiente de desenvolvimento DEV C++ e a segunda utilizando a plataforma Arduino na Atividade prática Ardumotic. Em termos de tempos letivos foram planeados 5 tempos de 45 minutos para cada uma, embora na prática, como descrito no ponto seguinte, tenham sido realizados 7 para cada, pelo acréscimo de um quarto dia com 4 blocos de 45 minutos.

A existência de duas fases com recursos diferentes, tem como objetivo principal complementar o método de ensino “tradicional” de programação, onde a

evolução da aprendizagem de conceitos é mais fácil de observar e avaliar taxologicamente, com uma abordagem ao currículo STEM permitindo aplicar os conhecimentos adquiridos dessa primeira parte numa finalidade concreta que promova a observação e contacto com aspetos inerentes à realidade, o estímulo do pensamento criativo e a capacidade de resolução de problemas.

4.4.2.1 Parte 1 – Aprendizagem na IDE DEV C++

Para iniciar a abordagem a esta temática é feita uma exposição, por vídeo-projeção, dos conteúdos e conceitos principais da temática Strings (Anexo B), passando pelos objetivos e a sequência de atividades a realizar, sempre que possível com interação formativa para avaliar o seu acompanhamento e perceção por parte dos alunos.

Propõem-se de seguida a resolução de exercícios práticos (Anexo C) no ambiente de desenvolvimento ao qual já estão mais ambientados, o DEV C++. Estes exercícios pretendem ser tipicamente enquadrados no que Newell & Simon (1972) situam no tipo de problemas “Moderadamente estruturados” da metodologia PBL.

Com estes exercícios apresentados numa ficha guiada pretende-se aplicar os conceitos que caracterizam as Strings e os métodos mais utilizados na sua manipulação. Esses pequenos desafios em forma de problema são do tipo “analisar o output da execução” ou “acrescentar ou alterar algumas partes de código para finalidades alternativas ou complementares”.

Concretamente, nesta fase de aprendizagem pretende-se que os alunos atinjam as seguintes competências:

- Saber distinguir entre uma variável simples e uma variável estruturada;

- Saber distinguir, assim como identificar semelhanças entre uma String e um Array de caracteres, através da sua criação, visualização/controlo da dimensão e acesso a todos os elementos;
- Reconhecer que nem todos os elementos da tabela ASCII são “caracteres” imprimíveis, particularmente o símbolo “\0”, contextualizando as suas funções relativamente a variável String;
- Conhecer a estrutura de armazenamento da variável String em memória e saber distinguir entre Strings e ponteiros para Strings, particularmente na utilização deste segundo em funções da sua manipulação;
- Constatar a necessidade de utilização de dados estruturados em Strings e saber utilizar as funções disponibilizadas para a sua manipulação na linguagem C para questões concretas.

4.4.2.2 Parte 2 – Atividade Ardumotic no Arduino

Nesta segunda parte, em referência aos padrões metodológicos adotados situa-se as propostas de aprendizagem por problemas apresentados no nível Moderadamente Estruturado, ou seja, com mais que uma solução aceitável, embora convergentes para o mesmo resultado final, e em que seja necessário procurar informação relativa aos conceitos e estrutura das Strings para implementar as soluções.

Pretende-se nesta fase que os conhecimentos previamente adquiridos sejam utilizados nos desafios propostos desta atividade prática e que o interesse dos alunos seja cativado pela observação de resultados para “além do Computador” e com uma utilidade concreta, mas com um grau de exigência de abstração gradual e acessível a todos os alunos. Desta forma, o guião da atividade, com os vários scripts nos seus passos sucessivos, foi estruturado de forma modular para permitir a aplicação dos

vários conceitos separadamente, permitindo ao aluno perceber e assimilar o seu funcionamento mais facilmente. Estas funcionalidades modulares são complementares, ou seja à medida que o aluno progride e as domina irá obter resultados mais completos integrando as diversas funcionalidades no mesmo programa.

Pretende-se com esta atividade, apresentada no documento fornecido aos alunos (Anexo I), orientá-los na aprendizagem e implementação, na plataforma Arduino, de um sistema de monitoramento de temperatura e humidade com visualização em display LCD de duas linhas, assim como através do Monitor-Série (MS) da IDE Arduino. Pretende-se também implementar funcionalidades que permitam interação com o sistema Arduino através do seu terminal MS que além de receber e visualizar informação de Debug permite também enviar, com base em Strings, ao próprio Arduino, permitindo o seu controlo em tempo real. Com esta atividade de aplicação prática num sistema físico, pretende-se transmitir aos alunos a percepção e consciência de que o poder computacional está diretamente relacionado com os dados e respetiva informação associada ao objeto que se pretende controlar.

Principais conceitos envolvidos:

- Criação e inicialização de Strings;
- Concatenação de Strings;
- Funções de pesquisa e manipulação de substrings;
- Comunicação via Serial COM.

Competências a adquirir pelo aluno:

- Saber aplicar os conhecimentos adquiridos num sistema digital autónomo com visualização de informação textual;

- Saber identificar e corrigir erros ou fazer alterações no sistema desenvolvido.

Material necessário (por grupo):

- Um computador funcional na secretária;
- Um Arduino Uno com cabo USB;
- Display LCD 16×2 (IIC I2C TWI Serial 1602 16×2);
- 7 Fios conectores MF;
- Módulo com sensores de temperatura e humidade - Shield DHT-11;
- Documentos impressos a disponibilizar por grupo:
 - Guião da atividade, (Anexo I);
 - Lista de funções sobre Strings, da sua estrutura e descrição breve, impressas em folhas A4, para serem utilizadas como referência e auxiliar de memória (Anexo K);
 - Tabela ASCII (Anexo J).

Na imagem seguinte representa-se a ligação completa do sistema, cujas conexões são as da tabela 9 com os respetivos pinos e funções:

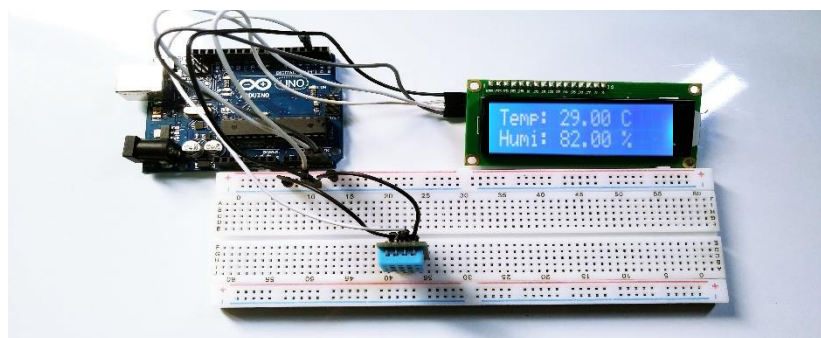


Figura 6 - Conjunto Arduino e Módulos, Display I2C e DHT11, com respetiva ligação.

Tabela 9 - Correspondência de ligações entre o Arduino e o Display I2C

GND	GND
VCC	5V
SDA	A4
SCL	A5

4.5. Descrição da operacionalização das aulas

Tal como descrito no ponto relativo ao planeamento das aulas, foi previsto existirem 10 blocos de 45 min (7,5 horas) disponíveis, para os dias 3, 5 e 10 de Março. No entanto devido a alguns atrasos no desenvolvimento da atividade, foi cordialidade cedido pela Professora cooperante, em acordo com o professor orientado, ter uma aula adicional, marcada para a quinta feita seguinte, acrescentando assim 4 blocos de 45min.

Nas descrições que se seguem, as referências a alunos concretos serão feitas por uma letra atribuída aleatoriamente, e que vai de A a M.

4.5.1. Descrição das aulas 1 a 3 – Primeiro dia

Data: Dia 3-3-2020 das 10h00-11h30 e das 11h45-12h30

Resumo: Apresentação; Introdução aos conceitos básicos; Início dos exercícios em DEV C++.

O início da primeira aula foi recetivo por parte dos alunos e com alguma naturalidade, pois já existia um bom conhecimento mútuo. Observou-se a falta do aluno C e da aluna N, sendo que esta última efetivamente não compareceu mais. Para formalizar a minha presença foi feita uma breve apresentação do enquadramento desta PES e dos seus objetivos.

De forma a aproveitar o tempo para preparar a apresentação inicial em PowerPoint e preparar a distribuição de recursos em papel, descritos a seguir, foi proposto aos alunos a realização de uma ficha de diagnóstico com duração de 30

minutos. Esta ficha teve como finalidade obter informação sobre o nível de conhecimentos e competências dos alunos relativamente aos conceitos dos 3 módulos anteriores. Na realização desta ficha de diagnóstico era esperado que os alunos respondessem nas folhas entregues em papel, mas como nem todos tinham caneta, ficou então decidido responder digitalmente no computador e cada um gravar o teste com o respetivo nome na pasta partilhada online, na qual já é habitual guardarem os seus documentos e trabalhos.

Durante a realização desta ficha manifestaram-se algumas dúvidas de interpretação a nível geral, e de conhecimentos específicos mais evidenciadas na temática Funções.

Após a finalização do diagnóstico foram entregues os documentos impressos necessários para as primeiras aulas, para serem seguidos durante a apresentação e servirem de recurso para tudo que se seguiria nas aulas seguintes:

- Manual de conceitos teóricos – Anexo H;
- Tabela ASCII – Anexo J;
- Ficha de exercícios a resolver em DEV C++ - Anexo C.

Iniciou-se a apresentação vídeo-expositiva (Anexo B) que decorreu até ao slide 6. Foram apresentados os conceitos principais a abordar, as atividades a realizar e a sua distribuição ao longo das aulas previstas. Na abordagem aos conceitos partiu-se da definição de um array ligando-o à definição de String como um tipo específico de array. Tirando partido do conhecimento prévio de terem sido utilizados drones programáveis nas atividades das últimas aulas, e que os alunos gostaram muito dessa atividade, foi utilizado esse exemplo para estimular alguma interação. Propôs-se imaginar que o Drone poderia ler periodicamente as altitudes medidas e guardar os

seus valores num array para análise posterior, e de imediato o aluno I respondeu que isso é possível através do sonar existente no Drone, e dialogámos sobre a possível forma dessa implementação e sobre o tipo de dados armazenados nesse array (inteiro, float ou outro).

Após os exemplos, demonstrou-se com recurso à tabela, disponibilizada em papel, a identificação de caracteres simbólicos e não simbólicos na tabela ASCII, particularmente o símbolo ‘\0’, passando desta forma à distinção entre Array e String. Descreveram-se as formas de inicialização de Strings e contagem de caracteres tendo em conta os índices dos elementos no array.

Os alunos demonstraram atenção e interesse durante a apresentação teórica e foram participativos. As observações desta aula constam no anexo G1, relativo a dimensão “Atitudes e Valores”, da avaliação sumativa. Pela interação com os alunos, aparentemente o conceito de String foi bem assimilado.

Após esta apresentação, foi pedido aos alunos para resolver os exercícios 1 e 2 da ficha de exercícios fornecida e que se encontra no anexo C, com respetiva resolução.

No primeiro exercício pretendia-se proporcionar o domínio do acesso aos caracteres individuais de uma String pelo seu índice de posição no array. Inicialmente os alunos demonstraram elevado nível empenho, no entanto cerca de metade manifestou algumas dificuldades, sendo que a outra metade demonstrava alguma autonomia. Uma das dificuldades evidenciadas foi a implementação e domínio do ciclo “for” e do ciclo “while”, no seu mecanismo de iteração, sendo que nestes primeiros exercícios estas competências eram necessárias para avançar.

Para ajudar neste tipo de dificuldades, e segundo Luxton-Reilly (2018) por ser uma ferramenta muito fértil no seu potencial de apoio à aprendizagem sobre a

execução de programas (apesar de poucos estudos conhecidos), no final do exercício 1 foi feita uma demonstração de utilização do depurador (Debug) do DEV c++, para observação do conteúdo das Strings, dos respectivos caracteres individuais e do valor de variáveis auxiliares durante a execução dos ciclos. Observou-se que esta funcionalidade não estava normal/operacional no computador dos alunos A e G, mas como se apresentava como anomalia da instalação do programa, não foi possível resolver.

Alguns alunos demonstraram uma motivação elevada, no sentido de querer ultrapassar as suas dificuldades, sendo que durante o intervalo o aluno L ficou com o professor para colocar e esclarecer dúvidas relacionadas com a implementação dos ciclos de controlo do exercício 1. O professor exemplificou de várias formas, usando o depurador como auxiliar para demonstrar o estado das variáveis de controlo durante a execução, assim como o conteúdo das variáveis do programa.

No exercício 2 apenas os alunos B e H conseguiram evidenciar a distinção entre as funções *scanf* e *fgetc* na sua forma de receção/criação de uma String via teclado.

4.5.1.1 Reflexão do primeiro dia

Das dificuldades manifestadas pelos alunos e pelo apoio prestado maioritariamente ao nível da implementação dos ciclos de execução e na interpretação do conteúdo atribuído à variável String associado às respetivas posições do array, considera-se que os exercícios do tipo problema geraram alguma dificuldade nos alunos apesar de as respostas esperadas serem curtas e quase diretas. Relacionado com essa dificuldade observou-se também que o hábito de consultar o manual fornecido foi praticamente inexistente, recorrendo-se ao improvisado e à “tentativa-erro” como alternativa.

Observando a reação dos alunos e a sua adaptação aos exercícios, constata-se que o tipo de exercícios elaborado não foi o mais bem-sucedido possível para uma fase inicial. A maioria dos alunos demonstrou dificuldades em entender a finalidade das perguntas, e observou-se também uma falta de domínio de alguns conhecimentos mais básicos como ciclos de controlo, o que dificultou a concretização dos objetivos das propostas. Por exemplo no exercício 1, o aluno H apresentou o seguinte script , em resposta ao pedido para apresentar sequencialmente todas as letras do nome introduzido:

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char** argv)
{
    char nome[10];
    printf("Escreva o seu primeiro nome: ");
    scanf("%s", nome);
    printf("\nA primeira letra do teu nome e:%c", nome[0]);
    for (int c=1; c<=9; c++)
    {
        printf("\nA primeira letra do teu nome e:%c", nome[c]);
    }
    return 0;
}
```

Ou seja, na linha dentro do ciclo *for*, não adaptou a String do *printf* de forma a descrever cada letra apresentada. Presume-se que apesar de ter alguma noção, o seu raciocínio não está completo quanto ao mecanismo de output do *printf*, assim como do estado de incrementação do ciclo *for* dentro das suas iterações.

No exercício 2, apenas um aluno efetivamente usou a função *strlen* como pedido na questão, e no entanto fê-lo com um script adaptado provavelmente obtido na internet:

```
int main ()
{
    char szInput[256];
    printf("Palavra: ");
```

```
    gets (szInput);  
    printf ("A palavra tem %u letras.\n", (unsigned)strlen(szInput));  
    return 0;  
}
```

Deduz-se desta forma que devia ter-se proporcionado um estímulo ao desenvolvimento cognitivo a partir de exercícios mais simples e diretos, para progressivamente, e a medida que assim se demonstre, se evoluísse para exercícios mais elaborados.

4.5.2. Descrição das aulas 4 a 7 - Segundo dia

Data: Dia 5-3-2020 das 10h00-11h30 e das 11h45-13h15

Resumo: Reflexão em grupo sobre as aprendizagens anteriores; Continuação da apresentação de conceitos sobre Strings; Continuação dos exercícios em DEV C++; Preenchimento do questionário investigativo da primeira parte da Intervenção.

A aula iniciou com a observação da presença de todos os alunos e com um breve diálogo formativo sobre pontos fortes e fracos das atividades da aula anterior, na perspetiva dos alunos, no enquadramento das metodologias utilizadas pelo professor e no contexto dos objetivos. Desse diálogo resultou que os exercícios com apresentação do tipo problema colocam um grau de dificuldade um pouco elevado para uma fase inicial desta temática, para o qual a maioria dos alunos não se sente preparado, referindo-se também que apesar de as respostas não serem complexas, teria de se adaptar a forma como são apresentados.

De seguida foi feita a apresentação dos slides sobre “Strings vs Ponteiros pra Strings”, Tendo em conta que algumas funções sobre Strings são utilizadas com ação direta nos seus parâmetros e outras retornam ponteiros para Strings.

Nesta apresentação foi sendo feito um encadeamento e demonstração direta com os respectivos exercícios relacionados, tal como previsto na sua preparação, não só para abordar a interpretação das questões colocadas nos exercícios, mas também para que da interação com os alunos se pudesse prever as dificuldades que a seguir se vinham a manifestar na resolução dos exercícios.

No exercício 3, a alínea a) considerou-se apenas para fim formativo, pois devido a um erro semântico na expressão `str1=str2`, identificado após algumas manifestações dos alunos ao obter um erro de compilação.

Na alínea “b)” a maioria dos alunos conseguiu apresentar verbalmente a diferença entre a String e o Ponteiro para String, no entanto algumas respostas foram nitidamente retiradas da internet, como por exemplo a resposta do aluno D:

“str_pointer1=str2; Com o ponteiro (str_pointer1) apontaras o endereço da memoria , mas diferente da str1=str2 usa-se diferente porque um ponteiro (str_pointer1) não pode ser igual a uma String porque têm funções diferentes.”

No exercício 4, sobre métodos de concatenação de Strings, apresentaram-se três formas diferentes de fazer concatenações para o mesmo caso, com grau de complexidade crescente e envolvendo concatenações múltiplas. O aluno E manifestou veementemente as suas dúvidas no raciocínio da concatenação, e com determinação para as ultrapassar deslocou-se ao quadro para expressá-las. Essa atitude foi muito positiva, porque cativou a atenção dos colegas, incentivando também a sua participação e raciocínio em grupo.

Concretamente, partindo da versão base a), em que se pretendia simplesmente juntar um sobrenome com um nome na mesma String, as versões b) e c) são duas formas diferentes de acrescentar um carater “espaço” entre os dois. Ou seja a

dificuldade dos alunos estava na percepção técnica/sintática do processo e da ordem em que se realiza esta concatenação:

```
printf("Escreva o seu primeiro nome: ");  
scanf("%s",nome);  
printf("Escreva o sobrenome: ");  
scanf("%s",sobrenome);  
strcat(nome," "); //concatenação do carater espaço à string nome  
strcat(nome,sobrenome); //concatenação da strins sobrenome à string nome  
printf("o nome completo\n");  
printf("%s\n", nome);
```

Explicando de várias formas diferentes conseguiu-se que a maioria dos alunos tivesse entendido e aplicado corretamente este método e a sua flexibilidade de utilização.

No exercício 5, sobre inversão de Strings, a alínea a), para verificação da existência de um palíndromo na palavra introduzida, foi desconsiderada/anulada devido à existência de um erro semântico na linha 13 do script apresentado, que não foi referido nas aulas, e ficou descrito na errata da ficha de exercícios. No entanto o aluno D (figura 7) conseguiu resolver parcialmente o pedido, apresentando um segundo processo para fazer inversão da String, e no qual apenas faltava a comparação de caracteres das Strings dentro do ciclo utilizando as suas coordenadas:

```

1  #include <iostream>
2  #include <cstring>
3  using namespace std;
4
5  int main()
6  {
7
8      char cstring[20];
9
10     cout<<"Palavra\n: ";
11     cin.getline(cstring,20);
12
13
14     int strSize = strlen(cstring);
15
16     for(int i=0;i<strSize*0.5;++i)
17     {
18         char ch = cstring[i];
19         cstring[i] = cstring[(strSize-1)-i];
20         cstring[(strSize-1)-i] = ch;
21     }
22
23
24     cout<<" : "<<cstring;
25
26     cout<<"\n\nEnter se tiver certo: ";
27     cin.get();
28
29     return 0;
30 }

```

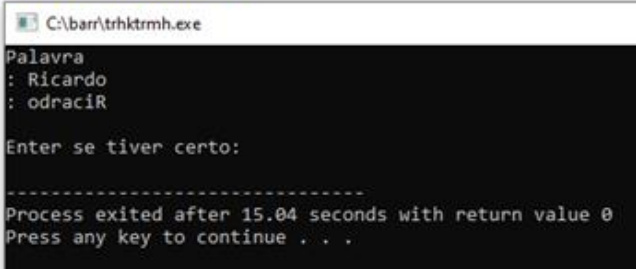


Figura 7 - PrintScreen do aluno D demonstrativo do exercício 5.a)

Um método alternativo, talvez mais simples teria sido a criação de uma segunda String, com o conteúdo invertido da primeira, como sugerido no enunciado e utilizar a função *strcmp()* para comparação das duas.

Na alínea b) do exercício 5, apenas o aluno D apresentou uma tentativa quase acertada de resolução, pois utiliza precisamente as funções de Strings sugeridas *strcmp()* e *strrev()*. No entanto o script apresentado foi nitidamente retirado da Internet e contém o código principal numa função faltando a principal *main()* e o pedido de introdução de uma palavra ao utilizador.

```

#include <iostream>
#include <cstring>
int ispalindrome(int n)
{
    char s[10], sr[10];
    itoa(n, s, 10);
    printf("%s", s);

```

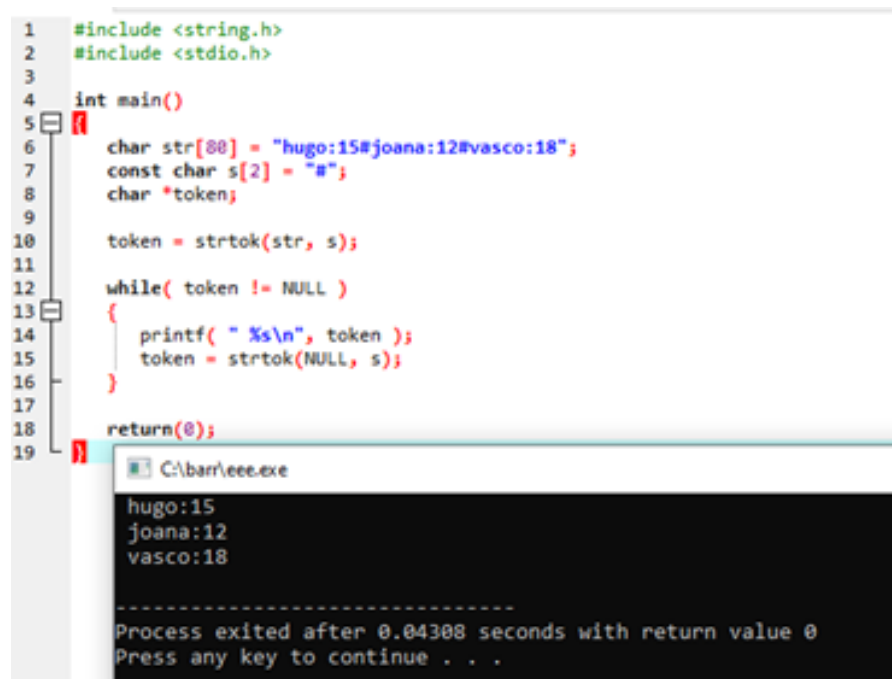


```

    sr = strrev(s);
    printf("\nReverso: %s", sr);
    if(strcmp(s, sr) == 0)
        return 1;
    else
        return 0;
}

```

No exercício 6, sobre Split de Strings, ao qual chegou apenas o aluno D (figura 8), e a par da sua demonstração de autonomia e interesse para entender o mecanismo desta função `strtok()`, o professor com uma pequena demonstração, e constatando que o aluno já tinha lido com atenção a pré-explicação colocada na ficha de exercícios, ajudou-o a perceber a aplicação do exemplo dado na questão colocada, mais propriamente o mecanismo e funcionamento do chamado “token separador”, que não é mais que um simples carácter (ou conjunto de caracteres) que o programador escolhe para subdividir a String principal.



```

1  #include <string.h>
2  #include <stdio.h>
3
4  int main()
5  {
6      char str[80] = "hugo:15#joana:12#vasco:18";
7      const char s[2] = "#";
8      char *token;
9
10     token = strtok(str, s);
11
12     while( token != NULL )
13     {
14         printf( " %s\n", token );
15         token = strtok(NULL, s);
16     }
17
18     return(0);
19 }

```

C:\bar\eee.exe

```

hugo:15
joana:12
vasco:18
-----
Process exited after 0.04308 seconds with return value 0
Press any key to continue . . .

```

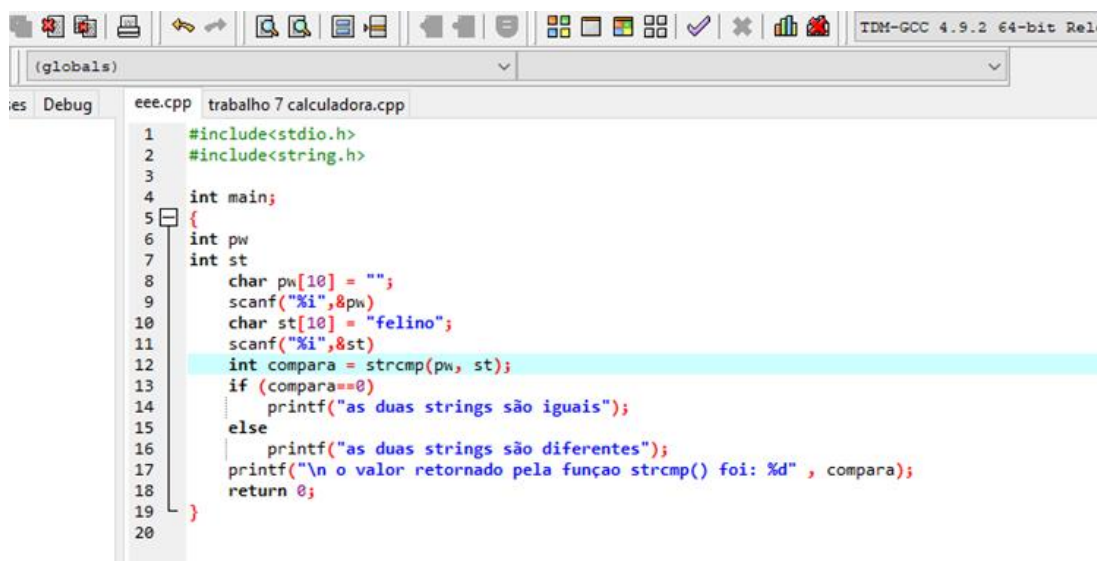
Figura 8 - PrintScreen do aluno D demonstrativo do exercício 6.a)

No exercício 6b), no qual é pedido para acrescentar o código necessário para contar o número de tokens, também apenas o aluno D o conseguiu realizar, embora de

forma incompleta na sua apresentação final, apenas com uma resposta textual. Pois apesar de ter entendido que seria necessário uma variável contadora dentro do ciclo já existente, faltou-lhe implementar a expressão para esse incremento iterativo. No entanto, pelo diálogo formativo com o professor ficou claro que o aluno tinha entendido a lógica do algoritmo apresentando a resposta seguinte:

“Tem que se meter int nt=0 e printf(“o numero de tokens é”,nt)”

No exercício 7, sobre comparação de Strings, também apenas o aluno D (figura 9) mostrou resultados, mas com pequenas falhas. Apesar de ter implementado corretamente a criação de uma das Strings para a variável “Password” e o seu input pelo utilizador, talvez não tenha entendido corretamente que a segunda String devia ter-se mantido com valor estático no programa:



```
1 #include<stdio.h>
2 #include<string.h>
3
4 int main;
5 {
6     int pw
7     int st
8     char pw[10] = "";
9     scanf("%i",&pw)
10    char st[10] = "felino";
11    scanf("%i",&st)
12    int compara = strcmp(pw, st);
13    if (compara==0)
14        printf("as duas strings são iguais");
15    else
16        printf("as duas strings são diferentes");
17    printf("\n o valor retornado pela função strcmp() foi: %d", compara);
18    return 0;
19 }
20
```

Figura 9 - PrintScreen do aluno D demonstrativo do exercício 7.

Devido ao atraso da aula anterior, optou-se por usar o tempo total deste dia para prestar apoio aos alunos na resolução dos exercícios em falta.

No final da aula foi preenchido pelos alunos um questionário investigativo sobre esta parte da Intervenção.

4.5.2.1 Reflexão do segundo dia

Com a metodologia baseada em problemas como predominante nos exercícios colocados aos alunos pretendeu-se estimular as competências cognitivas de raciocínio e autonomia na resolução de problemas através da aplicação dos conhecimentos adquiridos e recursos disponibilizados. Apesar de ter sido visível um maior esforço e empenho pelos alunos, surgiram dois aspetos evidentes a referir:

- Observa-se uma grande diferença de ritmos entre os alunos na resolução dos problemas;
- Por parte do professor surge a necessidade de um maior esforço no acompanhamento e apoio aos alunos, assim como a necessidade de uma estratégia adicional para sincronizar e definir patamares no ritmo global da turma.

Da análise do decorrer desta aula conclui-se que, apesar de o desempenho ter melhorado, a complexidade inicial dos exercícios pode provocar atrasos imprevistos. Um dos problemas identificados relacionado com este aspeto é a inexistência do hábito de consulta dos manuais fornecidos. Como hipótese de melhoria a este nível nesta temática, se numa fase inicial os exercícios propostos forem atómicos e simplificados, de forma a permitir a agregação contínua dos mesmos pelos alunos, é possível ir subindo o grau de complexidade controladamente.

Quanto à disponibilização da Internet aos alunos durante a realização dos exercícios, revela que existe a possibilidade de colocação de respostas literalmente

copiadas e pouco ou nada interpretadas, como na alínea 3b). Isto leva a crer que deve haver algum controlo deste recurso, de modo que a facilidade de acesso à informação não sacrifique o desenvolvimento e utilização da capacidade de raciocínio.

4.5.3. Descrição das aulas 8 a 10 – Terceiro dia

Data: Dia 10-3-2020 das 10h00-11h30 e das 11h45-12h30

Resumo: Apresentação da atividade prática Ardumotic; Montagem e testes da atividade; Início dos exercícios.

A aula iniciou com a verificação da presença de todos os alunos e com um breve diálogo de interação formativa quanto às dificuldades encontradas nas aulas anteriores. De seguida foi feita uma breve apresentação PPT, recapitulando conceitos essenciais das Strings, nomeadamente a inicialização e manipulação, passando pelas características que distinguem na prática as Strings de uma variável simples, e recapitulando as funções mais usadas para a sua manipulação, que seriam importantes na atividade prática que se seguiu.

O professor passou então à apresentação da atividade prática Ardumotic, partindo da descrição global de funcionamento e das ligações elétricas dos módulos. Para isso foi indicada a localização de todos os ficheiros de apoio necessários na pasta online:

- Manual da atividade prática, que contém o guião de montagem e testes, seguidos dos desafios para resolução baseados nos mesmos programas de teste, constante no Anexo I.
- Ficheiros com recursos específicos dos módulos utilizados, na pasta “Ardumotic Recursos”:
 - Manuais técnicos do display LCD I2C;

- Bibliotecas a inserir na IDE Arduino: do módulo DHT11 e do LCD I2C;
- A apresentação expositiva PPT das aulas, Anexo B;
- Ficha de exercícios da aula anterior com correção, Anexo C.

Foi feita a seleção de grupos através de sorteio. Para isso foi escrito o nome de cada aluno num pequeno papel fechado, e foi cordialmente pedido à Professora cooperante para fazer sua seleção e atribuição aleatória, Resultando 5 grupos de 2 alunos, e inevitavelmente um grupo de 3 alunos.

Após a distribuição do material da atividade, em que um kit estava já pre-montado para demonstração, os alunos começaram de imediato a sua montagem, guiada pelo ponto 2.1 e 2.2 do Guião, e a qual decorreu surpreendentemente rápido e praticamente sem problemas. Apenas o grupo VI necessitou de algum apoio nas ligações elétricas por faltarem alguns fios do tipo Macho-Macho e por isso o professor ajudou a identificar uma técnica alternativa de combinações com os disponíveis.

Na primeira fase de testes dos sensores de temperatura e humidade, guiada pelo ponto 2.3, o professor fez a demonstração da execução do Monitor Série do Arduino. A partir daí deixou aos alunos a execução do código exemplo deste ponto.

A maioria dos grupos conseguiu executar os programas e visualizar os resultados esperados à primeira. O grupo II encontrou uma dificuldade em importar as bibliotecas específicas dos módulos, mas com uma pequena orientação rapidamente resolveram esse detalhe.

Ainda na fase de testes, no ponto 5, inesperadamente todos os grupos estavam a constatar que a linha 2 do LCD não estava a apresentar o resultado esperado, apesar de as suas linhas de código serem exatamente idênticas às da linha 1.

Com alguma rapidez dois grupos (I e VI) conseguiram encontrar uma solução aceitável, embora não tenha sido identificado o erro propriamente dito durante a aula. Para isso, o professor recriou a situação após aula verificando que esse erro não se estava a repetir. Para confirmação, levou-se exatamente o mesmo programa que funcionou corretamente fora da aula, para executar num dos computadores da sala de aula. O erro voltou a acontecer, concluindo-se que seria devido a diferenças de compilador nas versões do Arduino instaladas.

Apresenta-se de seguida um exemplo de montagem completa, do grupo V, onde é possível observar o output em duas linhas independentes do LCD, neste caso com Strings personalizadas pelos alunos, e relacionado com o exercício do ponto 4:

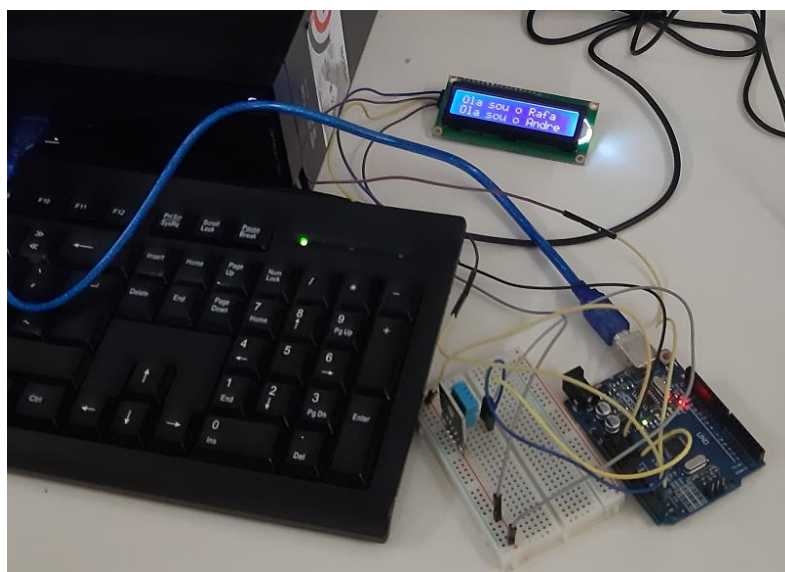


Figura 10 - *Exemplo de montagem e funcionamento do grupo V, com display nas duas linhas independentes, relacionado com o exercício do ponto 4.*

4.5.3.1 Reflexão do terceiro dia

Observou-se que a maioria dos grupos estavam já a realizar os exercícios do ponto 6, embora apenas dois grupos estivessem já a compreendê-lo, e tenha sido observado a tendência de improvisar em vez interpretar.

Quanto o problema da execução do script do ponto 5, o facto de os exercícios serem todos independentes, embora todos focados em finalidades complementares evitou que este “pequeno” problema tivesse provocado impedimentos nos restantes exercícios e na sua continuação.

Sendo que o grau de interesse e entrega à atividade foi elevado, e não sobrando tempo para o ultimo exercício (e mais interessante) assim como para o questionário investigativo final, foi cordialmente cedido pela professora cooperante em acordo com o professor orientado e na supervisão do Professor orientador/avaliador presente nesta aula, a possibilidade de estender a Intervenção em 4 blocos de 45 minutos correspondentes ao dia da quinta-feira seguinte (12 de Março) no horário da disciplina.

4.5.4. Descrição das aulas 11 a 14 – Quarto dia

Data: Dia 12-3-2020 das 10h00-11h30 e das 11h45-12h30

Resumo: Continuação da atividade prática Ardumotic; Teste teórico final; Preenchimento do questionário investigativo.

A aula iniciou com a confirmação da presença de todos os alunos e com a reatribuição do material completo da atividade da aula anterior aos grupos respetivos.

De modo a orientar a finalização da atividade, o professor observou diretamente com cada grupo qual o ponto de execução em que ficaram da aula anterior e as dificuldades.

Verificou-se que apenas dois grupos (I e III) estavam já no ponto 6 da atividade, embora sem ter ainda o resultado esperado.

No exercício 5: apenas o grupo IV conseguiu resolver. Os restantes tiveram dificuldades por falta de domínio de conhecimentos anteriores relativamente a definição de variáveis booleanas e atribuição valor.

Nos dois primeiros blocos de 45 minutos o professor prestou apoio aos alunos nas dificuldades encontradas, quando foi observado que todos os grupos tinham chegado ao ponto 6 do guião completou-se a apresentação em PowerPoint relativamente às funções de manipulação de Strings envolvidas, focando duas das mais complexas utilizadas no exercício do ponto 6:

- Função *strtok* – aplicada na receção de Strings via monitor Série com a finalidade de identificar comandos, para por exemplo alterar a mensagem de uma das linhas do LCD.
- Função *strncpy* – utilizada na função “*void lcd_msg_scrol_builder()*”, criada especificamente permitir um Scroll-Left, quando as Strings a visualizar no LCD são superiores a 16 caracteres (limite do LCD).

Neste ultimo ponto da atividade, apesar de todos os grupos terem conseguido implementar e executar pelo menos um comando via monitor-Série, apenas o grupo III conseguir implementar sem erros um segundo comando, para controlar o backlight do LCD (acender/apagar).

As principais dificuldades neste exercício foram:

- Assimilação do mecanismo da função *strtok* a partir do segundo token;
- O encadeamento das condições e seleção binária if/else encadeadas

A tendência para o improviso e a falta de hábito de consultar os manuais dificultou a progressão dos resultados desejados.

No terceiro bloco da aula foi pedido aos alunos para resolver uma pequena ficha de aferição de conhecimentos teóricos, com 6 perguntas de escolha múltipla.

De seguida foi pedido aos alunos para preencherem também um breve questionário investigativo online não identificado, idêntico e relacionado com o anterior.

4.5.4.1 Reflexão do quarto dia

De forma geral a colaboração dentro de cada grupo foi muito boa destacando-se os grupos IV e V pelo entusiasmo e entreajuda.

Observou-se com evidência que à medida que se ultrapassava obstáculos, o empenho dos alunos aumentava, sendo que o seu interesse e motivação foi crescente até ao final da atividade.

Como podemos observar nas fotos obtidas durante e no final desta aula, os alunos personalizavam mensagens a enviar para o respetivo display. Em alguns casos estavam ainda a tentar resolver bugs na sua implementação relacionados com a impressão completa das Strings, de forma a executar o Scroll-left.

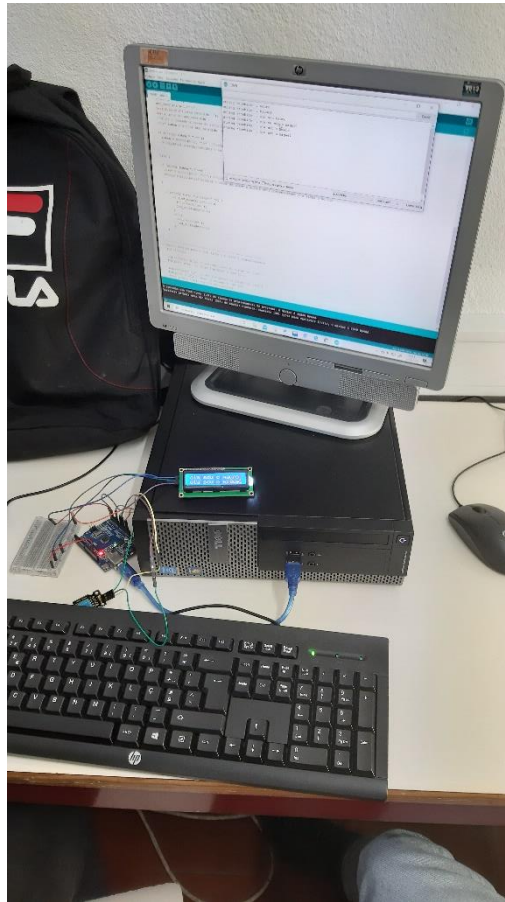


Figura 11 - *Exemplo de implementação do grupo VI.*

5. Avaliação da aprendizagem

Partindo da estratégia definida de divisão das aulas em duas partes e com os 2 respetivos recursos de aprendizagem, os critérios de avaliação foram definidos tendo em conta as observações possíveis e adequadas a cada uma. Desta maneira resultam duas avaliações, respetivamente uma para cada parte da Intervenção, e uma avaliação global relativa à evolução dos alunos em toda a experiência letiva desta Intervenção.

Na análise global da aprendizagem, pretende-se obter uma visão da sua evolução e da eficácias das respetivas estratégias, e não uma comparação direta dos recursos utilizados, dado que as suas atividades forma planeadas como complementares.

Desta forma consideram-se as dimensões relativas à aprendizagem dos alunos indicadas no Perfil do Aluno, descritas a seguir e as relativas à avaliação da própria Intervenção pedagógica.

5.1. Metodologia e instrumentos de avaliação das aprendizagens

Segundo o Decreto-Lei n.º 55/2018, “Enquanto processo regulador do ensino e da aprendizagem, a avaliação orienta o percurso escolar dos alunos e certifica as aprendizagens realizadas, nomeadamente os conhecimentos adquiridos, bem como as capacidades e atitudes desenvolvidas no âmbito das áreas de competências descritas no Perfil dos Alunos à Saída da Escolaridade Obrigatória”.

A avaliação dos alunos deve partir de um diagnóstico inicial e realizar-se continuamente até ao final das aulas através das componentes formativa e sumativa. Durante este processo de orientação e certificação, geralmente é difícil distinguir ou separar a avaliação formativa da sumativa.

A avaliação formativa pode ser vista como uma extensão do processo de ensino/aprendizagem, para o seu apoio e complemento. Com caráter contínuo e sistemático recorre a procedimentos, técnicas e instrumentos que permitam aferir a evolução de cada aluno, visando o ajustamento dos processos e estratégias utilizadas.

A avaliação sumativa traduz-se na formulação de um juízo global sobre as aprendizagens realizadas pelos alunos, tendo como objetivos a classificação e certificação final ponderada.

Nesta Intervenção, sendo o tipo de exercícios predominantemente à base de questões-problema, pretende-se considerar erro como algo positivo. Pois ele é a demonstração da execução de uma ideia mesmo que não funcionando (ainda). Valoriza-se sim, a atitude do aluno ao expor esse erro e a sua postura para encontrar uma solução.

“Partindo dos objetivos instrucionais cognitivos, atitudinais e de competências e da problemática inerente ao ensino de programação, há que ter bem definido o percurso e estratégias para atingir um nível cognitivo relativamente elevado, em raciocínio e abstração.” (Ferraz & Belhot, 2010). Entre essas estratégias está a forma de como lidar com os erros dos alunos, é necessário não denegrir o erro, mas sim transforma-lo em motivação para que o aluno veja as pequenas soluções como passos para o resultado global de um programa.

Pretende-se desta forma privilegiar a avaliação formativa, indispensável à orientação de todo o processo de ensino/aprendizagem.

5.2. Avaliação diagnóstica

A avaliação diagnóstica tem por finalidade aferir os conhecimentos dos alunos na fase inicial de um processo de aprendizagem. Através desta análise dos conhecimentos dos alunos é possível preparar o processo de ensino-aprendizagem a seguir.

É sempre mais fácil identificar as dificuldades dos alunos do que encontrar os processos ideais para os ajudar a ultrapassá-las, mas convém começar por essa identificação. Segundo Rocha et al. (2010), para um melhor diagnóstico das dificuldades, é necessário um processo sólido de avaliação, uma vez que este é o instrumento principal não somente para análise, mas, sobretudo para tomada de ações efetivas sobre as dificuldades verificadas. Não havendo a devida intervenção a partir do diagnóstico das dificuldades do aluno, não se está a praticar propriamente uma avaliação, mas uma condenação do aluno perante um provável fracasso, como se apenas ele fosse responsável.

Não só para avaliar, antes da Intervenção foi possível assistir a algumas aulas desta turma, o que me permitiu observar diretamente o desenvolvimento de cada aluno, interagir com eles, conhecer as suas personalidades e formar uma visão das suas capacidades e dificuldades. Não apenas para formalizar mas também para estruturar melhor essa visão foi elaborado um teste diagnóstico (Anexo A), para aferir com mais precisão esses conhecimentos nos temas já lecionados até então.

Nesse teste, o número de perguntas é naturalmente limitado, no entanto a sua centralidade na abrangência de conceitos, como apresentado na tabela seguinte, já permite constatar o limite inferior dos níveis de conhecimentos dos alunos.

Quanto à possibilidade de acesso à Internet, o qual foi permitido na expectativa de ser um recurso a ser utilizado para relembrar conhecimentos e conceitos, constatou-se que foi tendencialmente utilizada para procurar “respostas rápidas”, e é possível deduzir que várias foram diretamente “coladas” e não sendo mesmo adequadas, o que leva a presumir que seria mais benéfico para o estímulo do raciocínio ter sido inibido o acesso à internet durante este tipo de exercícios. Esta análise aponta para alguma dificuldade de raciocínio e hábitos de interpretação. Durante o teste apenas alguns alunos manifestaram dúvidas, sendo indicador que a maioria das vezes mesmo quando as têm, não são colocadas.

Na tabela seguinte, representando a descrição dos resultados quantitativos do Anexo A1, apresentam-se os conceitos abordados neste teste e as principais observações.

Tabela 10 – *Observações dos resultados da avaliação diagnóstica.*

Unidade	Conceitos /exemplos	Dificuldades observadas
Módulo 1	Fluxogramas, algoritmos (Pergunta 1): <ul style="list-style-type: none"> • Interpretação do fluxograma; • O operador %. 	Noções razoáveis dos conceitos básicos e de operadores em aproximadamente 50% dos alunos, e outra metade a demonstrar várias dificuldades.
Módulo 2	Mecanismos de controlo e execução (pergunta 2): <ul style="list-style-type: none"> • Ciclo <i>for</i> e <i>while</i>. 	De forma geral demonstram não dominam totalmente os ciclos <i>for</i> e <i>while</i> .
Módulo 3	Funções (Pergunta 3): <ul style="list-style-type: none"> • Identificação de Parâmetros; • Identificação de variáveis globais/loais. 	De forma geral demonstram pouco domínio da estrutura das funções.
	Conceito de recursividade de funções (Pergunta 4).	Apenas alguns alunos entendem o conceito de recursividade.

	<ul style="list-style-type: none"> • Definição; • Aplicabilidade. 	
--	---	--

Da análise das observações efetuadas, no módulo 1 e 2, constatando-se que algumas dificuldades correspondem a competências necessárias para a maioria dos exercícios, e tendo em conta que esta análise já foi feita entre o primeiro e segundo dia, considerou-se reforçar a avaliação formativa e o apoio aos alunos durante a resolução dos exercícios.

Quanto ao módulo 3, sobre programação estruturada com funções, como os seus conteúdos estavam previstos apenas para a atividade prática, de forma a minimizar as dificuldades e a complexidade dos exercícios, considerou-se adaptar as questões da atividade Ardumotic no que respeita a estes conceitos, e os programas com execução sequencial monobloco. Apesar de existir uma função indispensável no sketch principal - *lcd_msg_scrol_builder()* - foi já disponibilizada completa para utilização e interpretação apenas, como se pode observar no Anexo I (Manual-Guião da atividade prática), no problema do ponto 4.

5.3. Avaliação formativa

A avaliação deve ser um processo contínuo, e pode ser concretizada de varias formas, mas uma das principais finalidades é contribuir para a formação do aluno no seu processo e ritmo de aprendizagem. Esse ritmo é único em cada um, e por isso, a avaliação formativa serve para contribuir na sua autorregulação e para simplesmente transformar o erro num passo para sucesso.

A avaliação formativa é útil para o aluno assim como para o professor, servindo como meio de diagnóstico e adaptação de estratégias e processos de ensino, pois o professor também não é perfeito.

Nesta intervenção, sendo a disciplina de programação muitas vezes baseada na “tentativa-erro” depende muito do contexto e finalidade das atividades realizadas. Nomeadamente na metodologia PBL aqui adotada, a avaliação formativa tem um papel fundamental.

Assim, adotou-se como principais formas de avaliação formativa a observação e interação direta com os alunos, quer individualmente quer no grupo como turma.

A interação direta e individual com cada aluno, embora a mais exaustiva, permite constatar com precisão as suas dificuldades e ter uma resposta imediata que permita forma-lo, através da indicação das técnicas mentais adequadas, começando na formação do problema em causa, até ao acesso aos recursos que permitam chegar à solução pretendida.

Os diálogos em grupo também são muito importantes porque permitem o debate e reflexão sobre o método e tipo de exercícios utilizados pelo professor e identificar possíveis falhas na elaboração de enunciados, na tipologia dos exercícios, ou a forma da sua apresentação.

Para além da interação direta, o professor deve também usar grelhas de registo de observação individual dos alunos, para que seja possível em momentos fora da aula fazer análises e definir estratégias mais adequadas e específicas de medio/longo prazo. Os registos que foi possível efetuar encontram-se presentes nos vários anexos G, complementando a informação da avaliação sumativa.

5.4. Avaliação sumativa

A avaliação sumativa parte da definição de elementos observáveis e quantificáveis que se enquadram nas dimensões das Aprendizagem Essenciais (AE) a

atingir e estabelecidas. Essas AE, direcionadas aos Cursos Profissionais segundo Roldão et al. (2018) correspondem a Competências segundo o Perfil do Alunos à Saída da Escolaridade Obrigatória.

Na definição de Competências, como combinação de conhecimentos, capacidades e atitudes, considera-se que cada área curricular envolve essencialmente algumas Áreas de competências (AC). Selecionaram-se para a avaliação sumativa das aulas desta temática as AC que permitam uma correspondência a todos os elementos, dos três domínios, que contribuem para essas Competências Essenciais esperadas, como apresentado na tabela 11, onde se apresentam as respetivas ponderações definidas para as aulas dadas.

Para cada domínio apresenta-se também o conjunto de elementos específicos através dos quais se recolhe a informação útil.

Tabela 11 - *Componentes da avaliação sumativa baseada nas ACPA, com respetivos domínios, ponderação e elementos.*

Áreas de competências do perfil do aluno (ACPA)	Dimensão	Pêso (%)	Elementos
A - Linguagens e textos; C - Raciocínio e resolução de problemas; D – Pensamento crítico e criativo; I – Saber científico, técnico e tecnológico.	Conhecimento e capacidades cognitivas	50	<ul style="list-style-type: none"> • Exercícios/Atividades realizados; • Capacidade de resolver os exercícios/problemas; • Criatividade e atitude crítica;

<p>E – Relacionamento interpessoal;</p> <p>B – Informação e comunicação;</p> <p>F – Desenvolvimento pessoal e autonomia;</p>	Atitudes e valores	50	<ul style="list-style-type: none"> • Postura em sala de aula e participação nas apresentações; • Dinamismo em trabalho de grupo; • Empenho nas tarefas propostas; • Organização e método; • Iniciativa e autonomia
--	--------------------	----	---

De seguida são apresentados os resultados da avaliação sumativa por aluno e por dimensão, cujos valores foram obtidos em três momentos concretos da PES:

- No final da primeira parte, designada DEV C++,
- No final da segunda parte designada Arduino.
- No final da PES com o teste teórico de aferição;

Tabela 12 - *Classificação Global dos alunos por componentes de avaliação, das duas partes da PES.*

Aluno	Atitudes e valores		Conhecimento e Capacidades cognitivas			Nota Final
	DEV C++	Arduino	DEV C++	Arduino	T. aferição	
I	18	18,8	9,7	14,2	16,6	15,64
H	16,8	18,8	8,8	14,2	10,2	14,52
C	14,8	18,8	0	14,2	6,4	11,88
D	16,2	14,6	11,4	14,2	13,2	14,14
A	15,6	15,4	6,4	14,2	9,8	12,85
G	15,6	18,8	9,7	16	13,2	15,06
B	18,8	16,6	6,8	16	6,4	14,05
M	14,6	17,4	9,8	15	9,8	13,94
K	12,6	17,4	9,8	15	6,4	13,1
J	13,4	18,8	10,1	13	6,8	13,35
E	19,4	18,8	9,4	13	13,2	15,35
F	16,8	18,8	11,8	13	3,4	14,2
L	17,4	17,18462	12,4	13	13,2	15,04
Média 1	16,2	17,7	8,9	14,2	9,9	14,1
Média 2	16,9		11,0			

De uma forma global podemos também observar pelo gráfico seguinte os resultados sumativos nos dois momentos de avaliação principais.

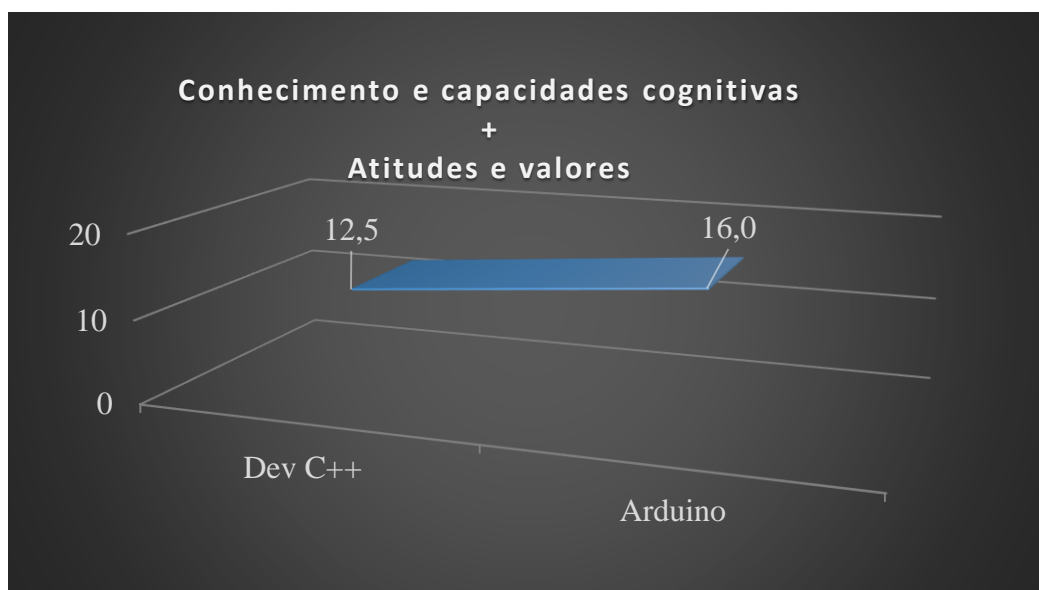


Figura 12 - *Evolução global da aprendizagem no conjunto das dimensões constituintes das Aprendizagens Essenciais.*

A primeira observação destes resultados indica que houve uma progressão positiva entre a primeira e a segunda parte da Intervenção, correspondente a uma melhoria quer na dimensão de Conhecimento e Capacidades Cognitivas quer na das Atitudes e Valores. A análise desta evolução é feita no capítulo seguinte, através da componente investigativa, em critérios definidos.

6. Componente investigativa da Intervenção

Nesta Intervenção, a dimensão investigativa é uma das partes mais importantes na perspectiva de análise e interpretação dos resultados obtidos. Sendo a educação um campo específico de investigação, ainda que no âmbito das ciências sociais, a Investigação Educacional permite-nos novos conhecimentos acerca do ensino, da aprendizagem e da administração educacional.

Tomando como referência o estudo de Luxton-Reilly et al. (2018), dos mais abrangentes até a data em termos de fatores de estudo observados, destaca-se a sua perspectiva global em que existem quatro entidades inter-relacionadas no processo de ensino-aprendizagem da Programação Introdutória, e que são o Aluno, o Ensino (que vai desde o professor aos recursos disponibilizados), o Currículo, e o Sistema de Avaliação. A análise de cada uma destas entidades advém de vários parâmetros partilhados entre si, e pretende-se aqui reutilizar essa análise dos parâmetros que com mais evidência é possível observar e caracterizar neste contexto e experiência concreta.

Sendo esta componente investigativa restrita ao “curto” período de tempo e conteúdos aqui abordados, com base na análise problemática do ensino de Programação e a avaliação diagnóstica inicial, surge então a necessidade de colocar uma questão/perspetiva de análise principal desta experiência letiva.

Essa questão centra-se na necessidade de interpretar a relação dos resultados obtidos na aprendizagem, assim como a sua evolução, com os fatores/dimensões possíveis de isolar com os quais esses resultados estão diretamente ligados, por representarem as escolhas pedagogias mais importantes, que são:

- A metodologia PBL adotada, focando a adequação das estratégias/atividades delineadas, na perspectiva da sua aplicabilidade e eficácia no ensino de

Programação desta forma, tendo em conta as características específicas de cada recurso;

- Os recursos pedagógicos utilizados nas duas partes da Intervenção, na sua adequação à temática, focando a diferenças específicas entre os dois e a sua influencia nos resultados e recetividade à aprendizagem.

Partindo dos resultados globais da avaliação sumativa da tabela 12, com a seguinte representação gráfica por dimensão das Aprendizagens Essenciais, faz-se nos pontos que se seguem a análise dos vários parâmetros que contribuíram para estes resultados, no enquadramento dos fatores referidos.

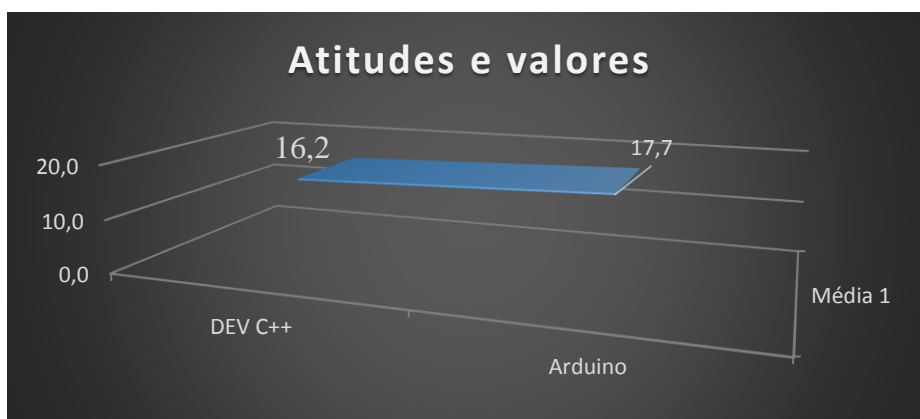


Figura 13 - *Evolução da componente Atitudes e Valores das AE*

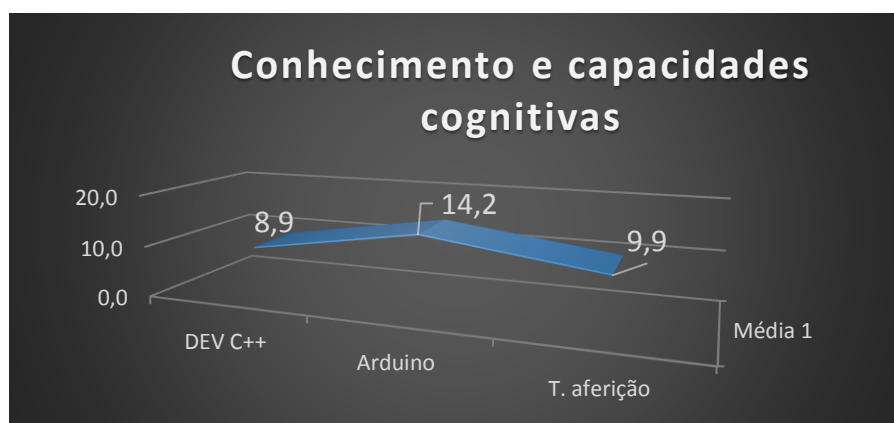


Figura 14 - *Evolução dos Conhecimentos e Capacidades Cognitivas das AE.*

Como fontes de informação foram utilizadas:

- As observações de aulas que permitiram avaliar as Atitudes e Valores relacionados com a temática;
- Resultados da avaliação sumativa dos Conhecimentos e Capacidades cognitivas, obtida pela correção dos exercícios resolvidos e respetivo código desenvolvido pelos alunos, que se encontra nos anexos G3, G4 e G5.
- Questionários de satisfação e avaliação da experiência de aprendizagem colocados aos alunos no final da primeira e da segunda parte da PES. Os questionários foram criados, disponibilizados e geridos através do Google Forms e encontram-se no anexo F.

Relativamente à análise e respetivas conclusões dos resultados dos questionários, tendo em conta que foram realizados em momentos relativamente distantes no tempo, propriamente 7 dias, deve ter-se em consideração que o fundo emocional das respostas pode ser diferente e que algumas questões têm conotação subjetiva. Ou seja, apesar de as questões serem análogas ou iguais, a tendência de resposta pode variar para além das razões relacionadas com os parâmetros das próprias aulas e respetivas atividades. Principalmente por esta razão, os resultados e conclusões desta componente investigativa não podem oferecer uma generalização ou aplicabilidade garantida por correspondência a outras experiências letivas.

Partindo desta informação disponível, organizada nos respetivos documentos, e da identificação dos fatores a analisar, faz-se a seguinte representação em diagrama de blocos para permitir a visão global da questão investigativa definida.

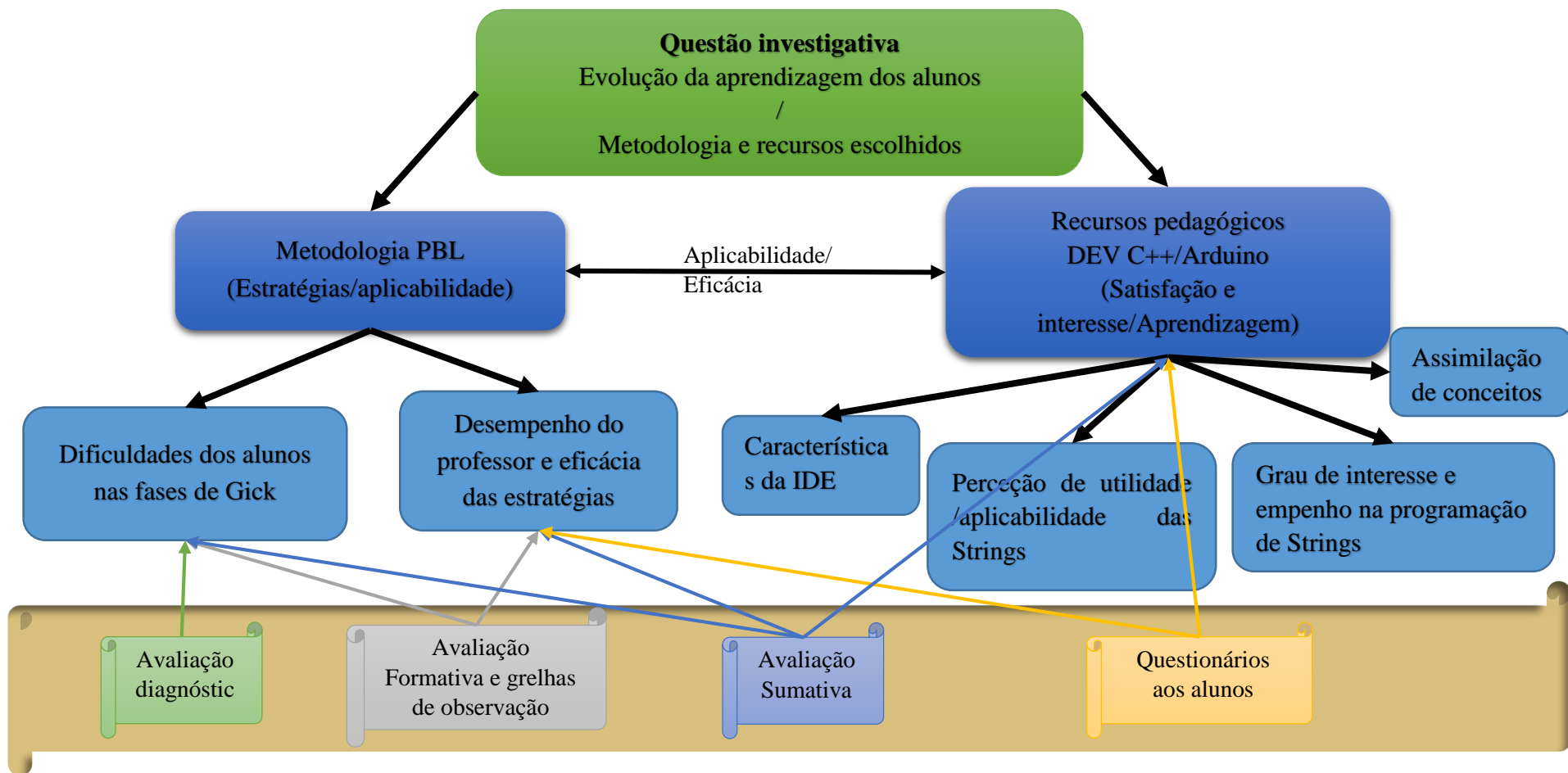


Figura 15 - Representação por blocos da Componente Investigativa da PES

6.1. Dimensão Investigativa «Metodologia»

Quanto à primeira dimensão da componente investigativa, a metodologia PBL adotada com respetivas estratégias e eficácia resultantes, utiliza-se como referência o modelo de Gick (1986). Embora seja relativamente antigo, a sua representação tem ainda como particularidade a sua semelhança de representação com o próprio conceito global de Programação.

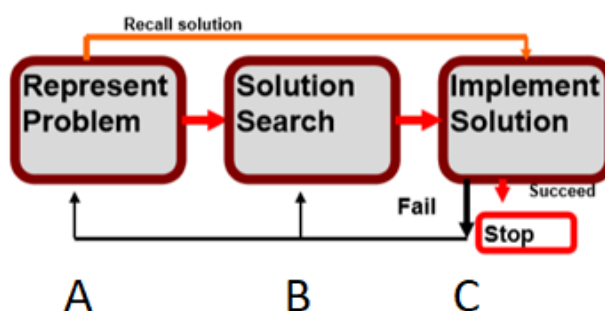


Figura 16 – *Representação da Metodologia de Aprendizagem por Problemas, modelo de Gick (1986)*

Tal como mostrado no diagrama global da questão investigativa, existem dois parâmetros de análise nesta dimensão:

- 1) Identificação da (s) fase (s), do modelo de Gick mais problemáticas no processo de aprendizagem dos alunos.**

Neste ponto relativo à metodologia adotada pretende-se analisar as principais dificuldades cognitivas dos alunos nas três etapas da resolução de problemas no modelo de Gick (1986) associando-as aos exercícios e atividades propostas da seguinte forma:

- a) Representação do problema: identificar variáveis, relações entre elas e mecanismos de controlo possíveis;

- b) Pesquisa da solução: definição de algoritmos possíveis ou identificação dos existentes para obter uma solução;
- c) Implementação da solução: Desenvolvimento de código utilizando conhecimentos já adquiridos para obtenção dos resultados desejados.

Através da análise da avaliação diagnóstica inicial e dos resultados que se seguiram na avaliação sumativa das três dimensões das AE, constata-se que as dificuldades mais evidentes se manifestaram nas fases B e C, respetivamente da Pesquisa da solução e da sua Implementação. Na fase B, as causas já apontadas, na descrição da operacionalização das aulas, são a falta de método de pesquisa de informação adequada e necessária para formar a correta visão técnica da solução dos problemas. A tendência para o imprevisto, levou frequentemente ao ciclo repetitivo e muitas vezes infrutífero de tentativa-erro. Na fase C, implementação das soluções, em reflexo da análise do diagnóstico inicial, verificou-se que uma das causas de dificuldade eram a falta de literacia de código, mais propriamente o que Luxton-Reilly et al. (2018, p. 6) designada por “capacidades de leitura e debugging de código”.

Luxton-Reilly refere que a leitura e interpretação de código complementa o processo de aprendizagem para as competências da sua implementação e desenvolvimento, pois o raciocínio é o mesmo e pode ser replicado. Observa-se na prática que, mesmo quando a finalidade da questão era perceptível e entendida, houve dificuldades na implementação de ciclos de controlo de execução iterativa e condicional.

Em termos de resultados de aprendizagem, conclui-se neste ponto que se a estratégia tivesse começado por reforçar e preparar melhor algumas bases dos conhecimentos dos alunos (assim como a capacidade de as utilizar), teriam sido

possíveis melhores resultados finais de aprendizagem, embora à partida isso implicasse uma redução de tempo disponível para atividade prática.

2) Eficácia da Metodologia quanto ao desempenho do professor e estratégias definidas.

Considerando o aluno como destinatário de toda a preparação pedagógica e como centro da análise, segundo Luxton-Reilly et al. (2018) deve ter-se em conta os fatores emocionais inerentes à experiência letiva, tais como a sua predisposição inicial, Interesse e empenho durante a aprendizagem, e sentimentos vividos durante a experiência.

Nesta perspetiva, utiliza-se como fonte de informação a opinião direta dos alunos através dos questionários que lhes foram propostos (Anexo F) e os resultados sumativos de avaliação, numa perspetiva de evolução entre as duas partes da PES.

Quanto à informação recolhida dos questionários, em reflexo da receptividade às atividades propostas e da satisfação na sua realização, filtram-se as 3 questões/opiniões seguintes:

a) Opinião geral sobre as aulas com DEV C++ e Arduino

No que respeita à satisfação sentida no envolvimento pelos alunos nas duas partes da Intervenção, e utilizando como fonte as respostas à seguinte questão, colocada nas duas partes da PES, obteve-se as respostas que se seguem:

Questão 3 do questionário 1: “As aulas desta Intervenção foram:”

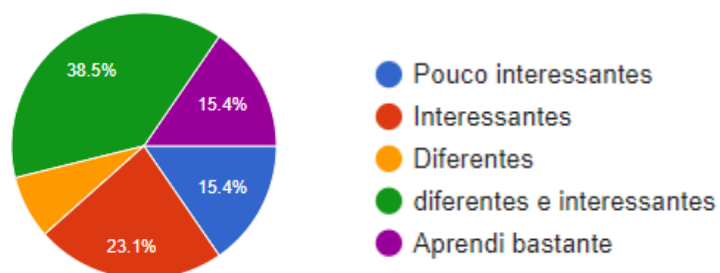


Figura 17 – Resposta à questão 3 do questionário 1, no final da primeira parte da PES

Questão 3 do questionário 2: “Esta atividade prática foi:”

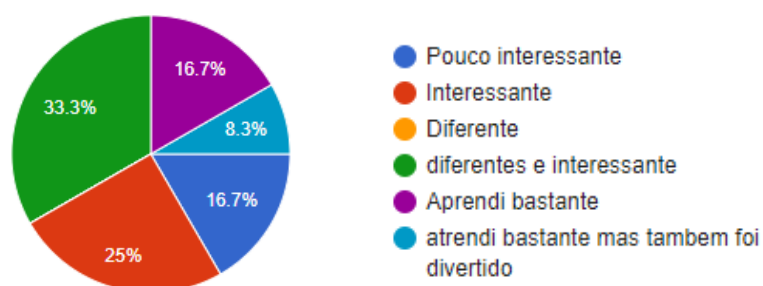


Figura 18 - Respostas à questão 3, no final da segunda parte da Intervenção

Agrupando as possibilidades de resposta a esta questão em 3 tipos que consigam abranger a percepção de uma experiência negativa, neutra ou positiva, em termos de aprendizagem, cativação e interesse do aluno, obtém-se a seguinte representação:

Tabela 13 - Grupos de respostas por tipo e com respetivo pêso na opinião global sobre as aulas nas duas partes da PES.

Tipo de resposta	Opções de resposta	DEV C++ (%)	Arduino (%)
Experiência de aprendizagem negativa ou neutra	<ul style="list-style-type: none"> • Pouco interessante; • Diferente. 	23	16,7
Aprendizagem positiva	<ul style="list-style-type: none"> • Aprendi bastante. 	15,4	16,7
Cativação e interesse positivos	<ul style="list-style-type: none"> • Interessante; • Diferente e interessante; • Aprendi bastante mas também foi divertido 	61,6	66,6

Observando a diferença entre a experiência positiva e a negativa ou neutra, constata-se que o fator de maior pêso foi na experiência positiva quanto à cativação e interesse, com aproximadamente 63% de média entre as duas partes da PES. Numa perspetiva de evolução e mudanças entre as duas fases, houve um aumento da perceção de experiência positiva na atividade com Arduino, distribuída entre o fator “Aprendizagem” e “Cativação e interesse”.

Relacionando esta análise com a melhoria dos resultados sumativos na segunda parte da intervenção, e com facto de o principal fator alterado ser o recurso educativo, pode afirmar-se que o aumento de interesse e motivação dos alunos na atividade Ardumotic teve influência no seu nível de empenho nas atividades e resultante melhoria na aprendizagem.

b) Nível de desempenho do professor e capacidade de resposta aos alunos

Relacionando a aplicabilidade e eficácia da metodologia PBL com o desempenho do professor e a qualidade das atividades propostas, obteve-se a seguinte representação gráfica resultante da questão “Nível das apresentações do professor e resposta aos alunos”, na figura 19.



Figura 19 - Respostas dos alunos ao desempenho do Professor nas duas partes da PES. Escala de resposta: 1-Discordo; 2-Neutro; 3-Concordo; 4 – Concordo plenamente.

Nesta questão deve referir-se um lapso na sua elaboração e apresentação no questionário 1 e 2. Era suposto ter sido aplicada uma escala de avaliação de Likert de 5 pontos, no entanto, no questionário 1, a opção em falta “discordo totalmente” estava com a designação “Column 5”. Neste caso decidiu-se ignorar as escolhas feitas pelos alunos nesta opção. No segundo questionário, constavam apenas 4 opções, faltando também a opção “Discordo totalmente”.

Este erro foi identificado durante a realização do questionário e informado aos alunos de forma a considerarem apenas as 4 opções corretas, no entanto, no questionário 1 houve ainda 4 escolhas da opção “Column 5”.

Desta forma, com a escala de pesquisa desta questão considera-se o valor 2 corresponde à opção “neutro” como o limite entre a opinião positiva e a negativa, e os resultados obtidos foram os seguintes graficamente.

Da análise de dados do gráfico podemos observar que, na perspectiva dos alunos o desempenho manteve-se idêntico nos vários parâmetros, assim como nas duas partes da PES, destacando-se pela evolução positiva o estímulo dos alunos, e pelo valor mais baixo a clareza e coerência das apresentações.

Numa visão global considera-se que a aplicação da metodologia PBL foi bem sucedida ao nível da adequação das atividades apresentadas aos conteúdos e do suporte necessário aos alunos, permitindo uma evolução das aprendizagens com significado e concretização de objetivos de Aprendizagens Essenciais. Deve destacar-se como fator determinante para estes resultados, a motivação e empenho associados ao melhor estímulo da aprendizagem revelado nesta questão.

c) Resposta livre de sugestão para as aulas de Programação

De uma maneira informal e aberta foi disponibilizada uma questão de resposta livre para possíveis sugestões para as aulas de Programação e obtiveram-se as seguintes respostas transcritas respetivamente dos dois questionários.

Questão: “Deixaria alguma sugestão para as aulas de Programação?”

Respostas obtidas no questionário 1, com DEV C++:

- *Precisa de explicar melhor a matéria*
- *Exemplos mais claros*
- *Mais divertidas e práticas*

- *Mais práticas e mais divertidas*

Respostas obtidas no questionário 2, com Arduino:

- *Foi bom*
- *Exemplos mais claros e diretos*

Da análise destas respostas pode efetuar-se duas constatações. Quanto à clareza e coerência das apresentações existe uma concordância entre as duas primeiras respostas do primeiro questionário e a percepção obtida na questão anterior relativamente ao mesmo aspeto, indicando que houve alguma dificuldade de interpretação dos pedidos propostos nas atividades.

Da segunda resposta do segundo questionário, considerando o mesmo tipo de parâmetro de análise, prevalece a dúvida se tem a mesma conotação que as duas referidas atrás ou não, pois da observação direta houve uma percepção de melhor facilidade na interpretação dos pedidos propostos, a par de um aumento significativo da percentagem de exercícios realizados.

Pelas duas últimas respostas do primeiro questionário, revelando a percepção dos alunos quanto ao interesse e entusiasmo nas atividades, e contracenando com a primeira resposta do segundo pode constatar-se uma mudança positiva. Neste caso é clara a diferença neste aspeto tendo em conta o recurso pedagógico, em que os exercícios típicos no IDE DEV C++ não foram vistos como “práticos” ou “divertidos” e apesar de haver apenas uma opinião no segundo questionário neste aspeto, existe uma concordância desta percepção com a observação direta das aulas e com o descrito no ponto anterior relativo à “eficácia da metodologia quanto ao desempenho do professor e estratégias definidas”.

6.2. Dimensão Investigativa «Recursos Pedagógicos»

Quanto à análise da evolução da aprendizagem dos alunos relativamente aos recursos utilizados, pretende-se avaliar o sucesso da utilização de cada recurso neste contexto específico, considerando-os complementares na estratégia global definida, porque além de serem de tipos diferentes, foram utilizados de forma sucessiva e com diferentes aplicações finais. Desta forma, é feita uma análise da evolução nas várias dimensões de avaliação sumativa e na perspetiva de satisfação/interesse de aprendizagem dos alunos com base nos respetivos resultados obtidos nos questionários.

6.2.1. Características específicas do Ambiente de Desenvolvimento

Neste ponto pretende-se observar a influência das características de cada Ambiente de Desenvolvimento (IDE) na lecionação desta temática e resultante aprendizagens. Dos questionários colocados aos alunos (Anexo F) filtraram-se as seguintes 5 perguntas com informação útil para esta dimensão investigativa. Algumas dessas questões são subdivididas em outras mais concretas.

Questão 1: “Qual o grau de dificuldade que sentiu no Ambiente de Desenvolvimento?”

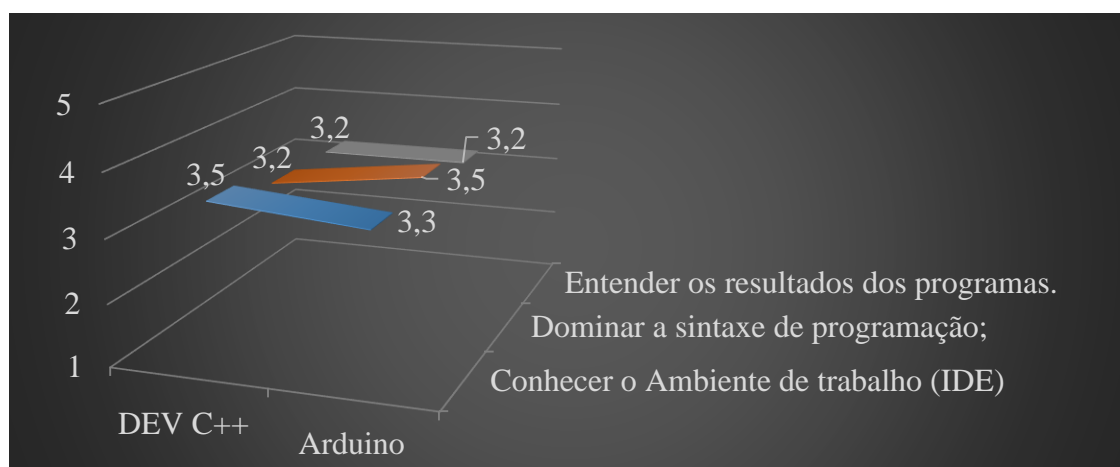


Figura 20 - Média das respostas obtidas sobre o grau de dificuldade na respectiva IDE de desenvolvimento. Escala de resposta: (Escala: 1-pouca; 5-muita).

Nesta questão, as respetivas respostas apontam para uma perceção global dos alunos equivalente perante ambos os ambientes de desenvolvimento, tendo sido a adaptação à IDE Arduino aparentemente fácil relativamente à do DEV C. Associando este facto com a observação direta das atividades com Arduino, quer da fase inicial de montagem e testes, quer da resolução dos respetivos exercícios, uma das razões terá sido o facto de o Arduino ser mais intuitivo e de as suas funcionalidades estarem associadas a um output observável, no entanto curiosamente o parâmetro “entender os resultados dos programas” teve exatamente o mesmo valor em ambos casos.

Quanto ao domínio da sintaxe percecionam-se mais dificuldades no Arduino, sendo a razão mais evidente a existência de uma quantidade maior de novas funções envolvidas, específicas dos módulos de hardware.

6.2.2. Dificuldades nos conceitos e percepção de utilidade e interesse

Nesta dimensão em que se pretende observar a influência da IDE da respetiva atividade nas dificuldades sentidas na aprendizagem da temática Strings consideram-se três parâmetros concretos como mostrado graficamente na seguinte questão:

Questão: “Nesta temática Strings considero:”

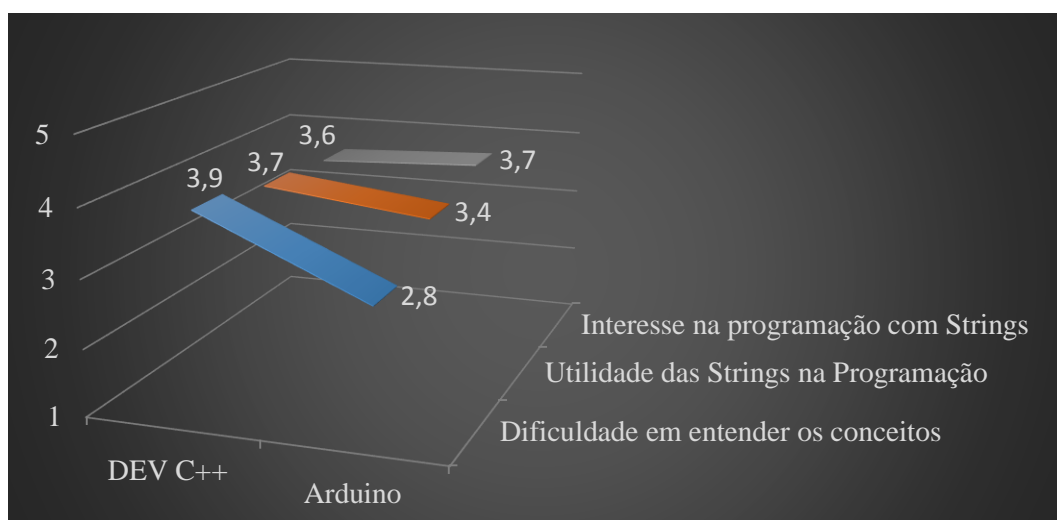


Figura 21 – Nível e evolução da percepção de utilidade, interesse e dificuldade na compreensão de conceitos. Escala de resposta: 1-Nenhuma; 5-Muita

Nesta dimensão podemos constatar que o grau da dificuldade manifestada na aprendizagem dos conceitos das Strings foi menor na plataforma Arduino, correspondendo também ao maior interesse nas atividades nessa plataforma. Quanto à percepção de utilidade das Strings, a comparação de resultados é contraditória com a questão seguinte, e considera-se que esta questão possa ter tido interpretações diferentes nos dois questionários.

6.2.3. Percepção de utilidade das Strings em aplicações reais

Nesta questão pretende-se observar a percepção dos alunos ao nível da aplicação concreta das Strings em finalidades que eles consigam (ou não) imaginar no seu dia-a-dia. Das respostas recolhidas obtemos a seguinte representação de evolução deste parâmetro:

Questão 2: “Qual o grau de utilidade que acha que as Strings podem ter na programação de coisas reais?”

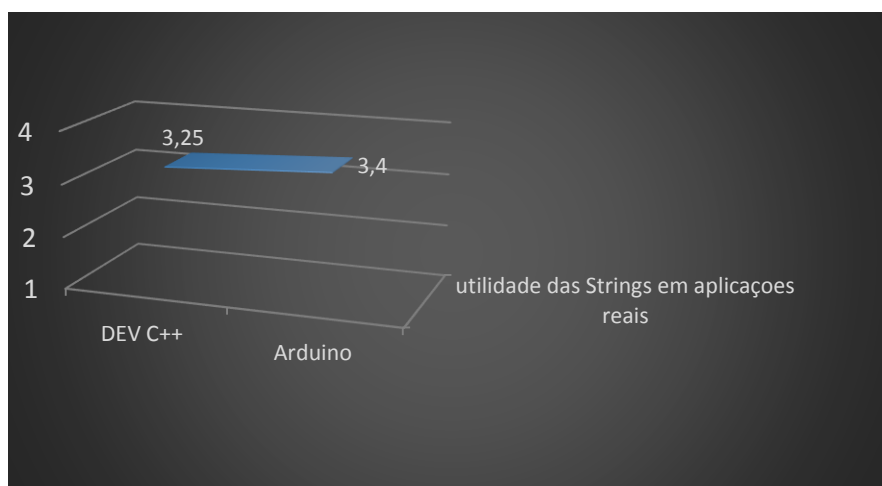


Figura 22 - *Percepção de Utilidade das Strings em Aplicações reais. Escala: 1 – Nenhuma; 4- Muita.*

Esta questão é complementar da anterior, mais propriamente do seu parâmetro “Utilidade da Strings na Programação” e permite argumentar a sua interpretação.

Neste caso, considera-se que “aplicações reais”, é mais fácil de associar a algo concreto como por exemplo a própria atividade Ardumotic, que proporcionou um conhecimento tácito dos conceitos das Strings, enquanto o termo “Programação” da questão anterior tem um sentido mais lato.

Desta forma, a representação gráfica anterior mostra uma tendência de subida desta percepção de utilidade das Strings, ao contrário do referido parâmetro na questão

anterior, e conclui-se que os alunos assimilaram a distinção entre um conjunto de conceitos, concretamente das Strings no seu contexto teórico de uma linguagem de Programação, e as formas/possibilidades da sua aplicação em finalidades concretas.

6.2.4. Cativação e interesse nos dois Ambientes de Desenvolvimento

Nesta questão apresentada apenas no segundo questionário, após a experiência com as duas Interfaces de Desenvolvimento, pretende-se objetivamente conhecer a opinião dos alunos quanto à cativação de interesse percebido nas duas fases da PES, considerando esse aspeto.

Questão: “É mais interessante aprender sobre Strings:”

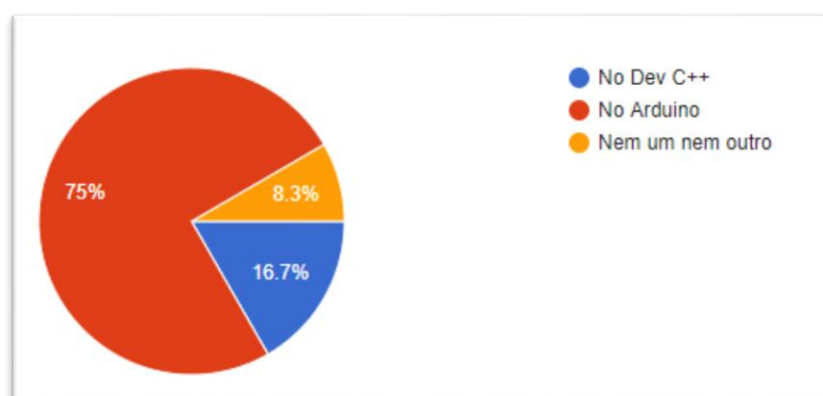


Figura 23 – Respostas sobre o interesse na aprendizagem sobre Strings, na IDE DEV C++ e Arduino respetivamente.

Da análise deste gráfico, constata-se que ao nível de cativação de interesse entre o Arduino e o DEV C++ existiu uma diferença de 75% para 16,7%, respetivamente, sendo os restantes 8,3% para a opinião “Nem um nem outro”. Estes valores, conjugados com a informação anterior são indicadores de que, se a estratégia for adequada aos conteúdos a lecionar, o Arduino tem um grande potencial a ser explorado, pois além de ter características naturalmente adequadas à aplicação da

metodologia PBL, tem a seu favor uma maior cativação de interesse e resultante empenho nas atividades.

6.3. Análise global

Observando de uma forma global as constatações da componente investigativa à luz dos resultados que permitiram caracterizar as suas dimensões – Metodologia e Recursos educativos, podemos tirar algumas conclusões como segue.

Na primeira parte da PES, com o recurso educativo DEV C++, sendo esta uma fase de abordagem inicial aos conceitos, pelos resultados de aprendizagem obtidos das várias fontes consideradas constata-se que o tipo de exercícios nele explorados deveriam ter sido mais básicos e diretos, ou seja focados na assimilação simplificada dos conceitos, e não de imediato com a sua imersão em desafios de resolução de problemas, mesmo que relativamente curtos e de resolução rápida. Porque a resolução de problemas presume a existência do domínio de ferramentas cognitivas resultantes de algum treino e domínio prévio de conceitos que servem de ponto de partida para o desenvolvimento da criatividade e capacidades de raciocínio que a resolução de problemas exige. Esta constatação resulta também da análise do primeiro ponto desta dimensão investigativa - Dificuldades dos alunos identificadas nas Fases de Gick (1986).

Na segunda fase da PES, constata-se que o nível de interesse e resultante empenho na atividade Ardumotic tenha contribuído para uma melhoria concreta nas aprendizagens. Além disso, e tendo em conta as competências esperadas da metodologia PBL, os vários indicadores aponta evidências de desenvolvimento da criatividade e capacidades cognitivas de resolução de problemas.

Identificando razões específicas que contribuíram para a melhoria dos resultados obtidos, para além das características inerentes dos próprios recursos educativos, deve focar-se pelo menos dois aspetos.

- O facto de a atividade Ardumotic ter sido baseado em exercícios relativamente melhor estruturados do que na fase do DEV C++ quanto à ponderação dos conhecimentos iniciais dos alunos no início da atividade, assim como a forma guiada, progressiva e modular da sua apresentação.
- Deve também reconhecer-se que os exercícios desta segunda atividade tiveram relativamente menor grau de dificuldade, por serem baseados na replicação de código já existente, alterando apenas variáveis análogas, e na observação da execução do programa base fornecido, o qual permitia uma referência para completar os resultados pretendidos.

Comparativamente, quer o DEV C++ quer o Arduino, e isolando os pontos fortes deste segundo na metodologia PBL, conclui-se que ambos podem ser usados em qualquer fase de aprendizagem, requerendo no entanto e da mesma forma, que o grau de complexidade das atividades nas suas fases concretas seja proporcional ao ritmo de aprendizagem esperado dos alunos, e ao nível de domínio de conceitos necessário em cada fase.

7. Reflexão final

Nesta reflexão final pretende-se apresentar uma visão global quer dos aspetos pedagógicos descrita neste relatório, quer da experiência pessoal vivida da qual resultou. Implicitamente apresenta-se uma breve descrição do meu percurso na profissão docente até à data, e das motivações que precedem este Mestrado Profissionalizante.

A minha experiência com o Ensino de Informática começou em 2006 como docente contratado no concelho de Sintra, lecionando a disciplina de TIC. Nos anos seguintes, até 2011 lecionei em escolas dos vários conselhos de Lisboa - Odivelas, Amadora e Oeiras - tendo sido a maior parte da experiência letiva em Cursos Profissionais nas disciplinas de Redes e Programação.

Nessa experiência profissional, e por felicidade das oportunidades, o facto de ter conhecido vários tipos de ambiente escolar, específicos de cada área geográfica e respetivo ambiente social, foi intencional. Essa possibilidade permitiu-me contracenar com uma grande diversidade de perfis de alunos e micro-culturas, enriquecendo não só as minhas competências letivas mas também as sociais. Paralelamente a minha perceção da escola como organização educativa também foi mudando, à medida que me apercebia das várias dimensões que a constituem e da importância do conhecimento do Currículo e da liberdade necessária de ação do professor como produtor de cultura e conhecimento.

Sobre essa liberdade não podia deixar de me aperceber também da responsabilidade e grau de exigência exponencial para a melhoria efetiva da “arte de ensinar”. Por não ser apenas a quantidade de palavras que determina a quantidade de

conhecimento transmitido, mas sim a assertividade de estratégias e a eficácia de utilização dos recursos disponíveis.

Considerando a indissociabilidade do lado físico e mental do ser humano, e segundo António Damásio, o Homem evolui para conciliar a emoção com a razão, (Grupo Abril, 2016). O professor é o principal promotor das experiências que irão proporcionar as condições ideais para esse processo. Mais do que aprender a executar uma tarefa, dominar a inter-relação de conceitos ou alcançar uma avaliação positiva, é necessário (e também inevitável) viver emocionalmente essa experiência, atribuir-lhe significado, integradamente na evolução social e cultural. Nesse sentido acredito que o professor deve ter uma visão pessoal na sua especialidade, que contribui para uma ligação com os seus alunos pela partilha da experiência de ensino/aprendizagem em todos os seus horizontes e potencialidades.

Até a data da realização do relatório desta Intervenção passei por outras experiências profissionais, quer no desenvolvimento de software e hardware para plataformas, quer comerciais quer pessoais em projetos investigativos e lúdicos. Essa experiência, quase sempre envolvendo Programação, direta ou indiretamente, permitiu-me constatar que a resolução de problemas e o pensamento criativo estão ligados, e deve considerar-se as várias dimensões da condição humana no seu equilíbrio emocional e físico com o ambiente que o rodeia, não só pelo discernimento necessário para inferir soluções mas para que a direção da criatividade seja para equilíbrio com esse ambiente.

Mas para ensinar não basta saber e dominar os conceitos. A par da proposição de Luxton Reilly et al. (2018), em que "as dificuldades dos iniciantes de programação podem ser consequência de expectativas irrealistas e não da complexidade intrínseca

da matéria lecionada”, podemos tirar duas interpretações da realização desta PES. Uma ligada aos problemas observados na primeira parte da intervenção, onde se concluiu que a complexidade da apresentação dos exercícios, apesar da objetividade de respostas esperadas, colocou dificuldades na interpretação e raciocínio aos alunos. E por outro lado, associando essa visão à segunda parte da PES, com a atividade Ardumotic, pode afirmar-se que a motivação e a forma de aprender os conceitos, resultante dos recursos utilizados, pode efetivamente fazer a diferença nos resultados. Ou seja, mesmo que a complexidade seja relativamente elevada (considerando a extensão da atividade prática e a quantidade de conceitos envolvidos), se a determinação dos alunos for também elevada consegue-se promover a sua autonomia e empenho proporcionalmente.

Segundo Leppan (2007), considerando que desenvolver um programa pode ser descrito como a criação de planos para resolver um determinado problema, o foco está na capacidade de descoberta desses planos, pois a parte restante é apenas a utilização de um recurso que os implementa. Este ponto de vista e a sua associação com as capacidades cognitivas concretas dos alunos deve ser tido em conta na decisão estratégica do tipo de exercícios a apresentar.

Nesta linha de raciocínio, os exercícios utilizados nesta PES tiveram como grande componente de competências de programação a capacidade de leitura e interpretação de código, sendo que as respetivas soluções derivam dessa interpretação para que fosse possível completar com os scripts em falta. Dessa forma, sendo que essa competência é precisamente um dos pontos fracos típicos, não só desta turma mas em geral nesta fase de aprendizagem, observaram-se dificuldades na progressão e qualidade esperada nos resultados sumativos.

Quanto à eficácia da Metodologia de aprendizagem por problemas, conclui-se que o resultado foi muito positivo, tendo conseguido a sua aplicação final com eficácia nos recursos educativos utilizados, e inferido conhecimento sobre o grau de complexidade e tipo de estrutura necessário nos cenários de aprendizagem para que isso fosse possível.

Focando os aspetos em comum entre o Pensamento Computacional e a Transdisciplinaridade, nesta fase atual da nossa realidade, a evolução acelerada das formas e meios de comunicação, e de aplicação das metodologias de ensino reflete-se também na forma de interação entre o professor e o aluno.

A par disso, existe já uma boa maturidade nos recursos pedagógicos disponíveis que torna muito interessando e motivador a sua utilização. Inevitavelmente, a flexibilidade esperada do professor é também muito grande, sendo-lhe exigido uma apurada visão sobre o conceito de Competências, e da potencialidade dos recursos escolhidos. Dessa forma é possível transformar com entusiasmo as atividades letivas em experiências construtivas e significativas para os alunos.

Nesta perspetiva final, mas olhando para trás, para o ponto de partida de todo este processo que envolveu a Intervenção, não podia deixar de colocar uma questão. O que foi escolhido primeiro, a Metodologia ou o Recurso educativo? Na prática ambos fazem parte do mesmo, do processo de proporcionar conhecimentos e capacidades cognitivas transformadas em competências. E mesmo que pelas razões pessoais que inevitavelmente qualquer docente terá haja uma tendência, apenas em conjunto fazem sentido, e é sempre necessário conhecer todas as dimensões que caracterizam o enquadramento pedagógico para que seja possível tornar útil a combinação entre os dois.

Referências

- Direção geral da Educação. (2020). *Iniciação à Programação 1.º Ciclo*. Obtido de Direção Geral da Educação : <https://www.dge.mec.pt/iniciacao-programacao-1o-ciclo>
- Agência Nacional para a Qualificação e o Ensino. (2019). *Cursos Profissionais*. Obtido de <https://anqep.gov.pt/np4/36.html>
- Agrupamento de Escolas Eduardo Gageiro. (2015). *Projeto Educativo 2015 / 2019*. Obtido de <http://www.eduga.pt/>
- Agrupamento de Escolas Eduardo Gageiro. (2020). *Cursos Profissionais* . Obtido de <http://www.eduga.pt/>
- Arduino AG. (2020). *Arduino*. Obtido de <https://www.arduino.cc/>
- AVR-Libc. (2020). *Standard C library for AVR-GCC*. Obtido de https://www.nongnu.org/avr-libc/user-manual/group__avr__string.html
- Beghetto, R., & Kaufman, J. (2009). *Beyond Big and Little: The Four C Model of Creativity*.
- Bogliolo, A. (2020). Clementoni - Robomaker, Starter.
- Dale, R. (2008). Construir a Europa através de um Espaço Europeu de Educação. *Revista Lusófona da Educação*.
- Damas, L. (1999). *Linguagem C – 24ª Edição*. FCA.
- Direção Geral da Educação. (2017). *Perfil dos Alunos À Saida da Escolaridade Obrigatória*. Obtido de Direção Geral da Educação: https://www.dge.mec.pt/sites/default/files/Curriculo/Projeto_Autonomia_e_Flexibilidade/perfil_dos_alunos.pdf
- Direção Geral de Formação Vocacional. (2005). Programa da disciplina Programação de Sistemas de Informação.
- Direção-Geral da Educação. (2020). *Programa Territórios Educativos de Intervenção Prioritária*.
- El-Abd , M. (2017). A Review of Embedded Systems Education in the Arduino Age: Lessons Learned and Future Directions.
- Ferraz, A., & Belhot, R. (2010). *Taxonomia de Bloom: revisão teórica e apresentação das adequações do instrumento para definição de objetivos instrucionais*.
- Ferreira, I., & Teixeira, A. (2010). Territórios Educativos de Intervenção Prioritária. *Sociologia: Revista do Departamento de Sociologia da FLUP*.
- Gick, M. (1986). Problem-Solving Strategies.
- Gomes, A., Henriques, J., & Mendes, A. (2008). Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de computadores.
- Grupo Abril. (2016). *Entrevista a VEJA*. Obtido de Abril.

- Hoffer, B. (2012). *Satisfying STEM Education Using the Arduino*.
- jamal, S., Ibrahim, N., Halim, N., & Osman, J. (2020). *A Systematic Review Analysis: Problem Based Learning On Creativity Domain In Problem Solving For Secondary School*.
- Jones, E. (2016). The trouble with computational thinking.
- Kirkley, J. (2003). *Principles for Teaching Problem Solving*.
- Leppan, R., Cilliers, C., & Taljaard, M. (2007). Supporting CS1 with a Program Beacon Recognition Tool.
- Luxton-Reilly, A., Becker, B., Ott, L., & Sheard, J. (2018). Introductory Programming: A Systematic Literature Review.
- Newell, A., & Simon, A. (1972). *Human Problem Solving*.
- Omar, S., Harun, J., halim, N. D., Surif, J., & Muhammad, S. (2017). Investigating the level of scientific creativity of science students. *Advanced Science Letters*.
- Petit, A. (2006). *School and the Production of Society*.
- Piedade et al., J. (12 de julho de 2019). A Cross-analysis of Block-based and Visual Programming Apps with Computer Science Student-Teachers.
- Prensky, M. (2001). Digital Natives, Digital Immigrants.
- Price, B., Hirst, A., Johnson, J., Petre, M., & Richards, M. (2002). Using Robotics for Teaching Computing Science, and Engeneering at a Distance.
- Ritchie, B. W. (1988). *The C Programming Language - Second Edition*. Prentice Hall.
- Rocha, P. S., Ferreira, B., Nunes, D. S., & Góes, H. C. (2010). Ensino e Aprendizagem de Programação: Análise da Aplicação de Proposta Metodológica Baseada no Sistema Personalizado de Ensino. *Novas Tecnologias na Educação*.
- Roldão, M., Peralta, H., Martins, I., & Orvalho, L. (2018). *PARA A CONSTRUÇÃO DE APRENDIZAGENS ESSENCIAIS BASEADAS NO PERFIL DOS ALUNOS*.
- Rubio, M., Hierro, C., & Pablo, Á. (2013). *USING ARDUINO TO ENHANCE COMPUTER PROGRAMMING*.
- Santos, E., Fernandes, E., & Fermé, E. (2005). Utilização de Robots no ensino de programação: o projeto droide.
- Santos, T. A. (2020). *Teresa e Alexandre Soares dos Santos - Iniciativa Educação*. Obtido de Iniciativa Educação: <https://www.iniciativaeducacao.org/pt>
- SOUZA, N., SIBILA, M., & PUNHAGUI, G. (2013). *Superando o erro como fracasso na construção de uma avaliação formativa*.
- Steinberg, A. (1997). Real Learning, Real Work.
- Tedre, M. (2016). The Long Quest for Computational Thinking.
- Universidade do Porto - Faculdade de engenharia. (2020). *TABELA ASCII*. Obtido de https://sigarra.up.pt/feup/pt/web_page.inicial

Usta, E., & Akkanat, C. (2015). Investigating scientific creativity level of seventh grade students. *Procedia - Social and Behavioral Sciences*.

Legislação Consultada

Decreto-lei nº74/2004 do Ministério da Educação (2004). Diário da República: Série I-A, n.º 73. <https://dre.pt/application/file/a/210759>

Portaria n.º 550-C/2004 do Ministério da Educação (2004). Diário da República: Série I-B, n.º 119. <https://dre.pt/application/file/a/238887>

Portaria 916/2005 do Ministério da Educação (2005). Diário da República: Série I, n.º 185. <https://dre.pt/application/file/a/147708>

Despacho n.º 6478/2017 do Secretário de Estado da Educação-Educação (2017). Diário da República: II Série, n.º 14. <https://dre.pt/application/file/a/107756793>

Despacho n.º 6494/2018 do Instituto Politécnico de Lisboa - Instituto Superior de Engenharia de Lisboa (2018). Diário da República: Série II, n.º 125. <https://dre.pt/application/file/a/115620238>

Decreto-lei nº55/2018 da Presidência do Conselho de Ministros (2018). Diário da República: Série I, n.º 129. <https://dre.pt/application/file/a/115648908>

Anexos

Anexo A - Teste diagnóstico

Anexo B – Apresentação Expositiva de conceitos em PowerPoint

Anexo C – Ficha de exercícios DEV C++

Anexo D - Teste de Aferição

Anexo E1 - Planificações das aulas do dia 03/03/20

Anexo E2 - Planificações das aulas do dia 05/03/20

Anexo E3 - Planificações das aulas do dia 10/03/20

Anexo E4 - Planificações das aulas do dia 12/03/20

Anexo F – Respostas aos questionários investigativos

Anexo G1 – Avaliação sumativa – Dimensão Atitudes e valores no DEV C++

Anexo G2 – Avaliação sumativa – Dimensão Atitudes e valores no Arduino

Anexo G3 – Avaliação sumativa – Dimensão Conhecimento e capacidades Cognitivas
DEV C++

Anexo G4 – Avaliação sumativa – Dimensão Conhecimento e capacidades cognitivas
Arduino

Anexo G5 – Avaliação sumativa – Dimensão Conhecimento e capacidades cognitivas
Teste Final

Anexo G6 – Avaliação sumativa – Resultado global

Anexo H – Manual das aulas teóricas iniciais

Anexo I – Manual e Guião da atividade Ardumotic

Anexo J – Tabela ASCII

Anexo A - Teste diagnóstico

Programação e Sistemas de Informação - 10º Ano Data: 03/03/2020

Módulo 4 – Estruturas de Dados Estáticas - Strings

Professor: Paulo Ferreira (Intervenção supervisionada)

Aluno

Nome: _____ N.º: _____

Através das seguintes questões pretende-se avaliar de forma resumida os conhecimentos do aluno em Programação antes da Intervenção Pedagógica.

1. Observe o algoritmo seguinte.
 - a) Complete a palavra que falta nos retângulos a vermelho:

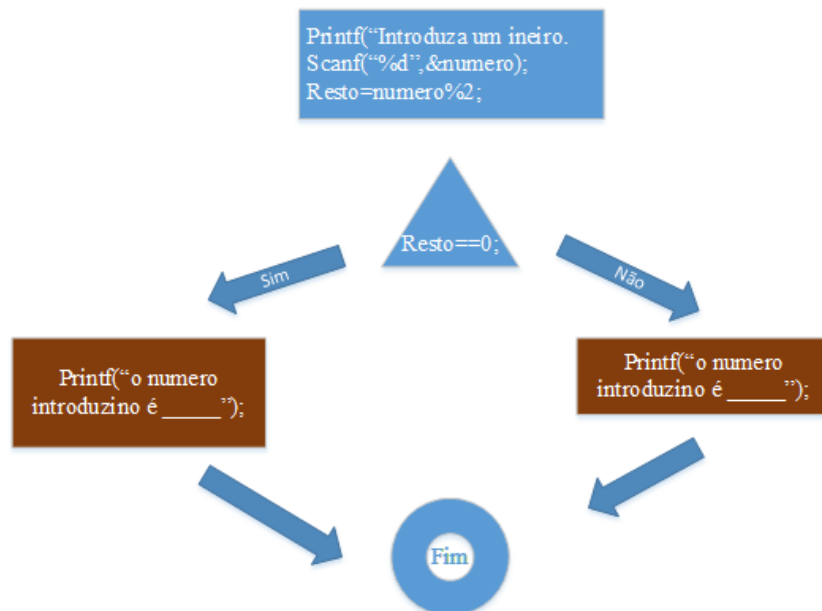


Figura A- 1 – Algoritmo

- b) O que entende por algoritmo?
2. Observe o seguinte programa.
 - a) Qual o output da sua execução?

```
void main()
{
    int j, soma = 0;
    for (j = 1; j <= 10; j++)
    {
        soma = soma + j;
        printf("%d ", j);
    }
}
```

```

    }
    printf("\nA soma é : %d\n", sum);
}

```

- b) Para que pudesse usar o ciclo While como alternativa, substitua o “???????” no programa seguinte pela linha de código correta para que o resultado da execução seja o mesmo do caso anterior .

```

void main()
{
    int j, soma = 0;

    while (j<=10){
        soma = soma + j;
        printf("%d ",j);
        ????????;
    }
    printf("\nA soma é : %d\n", soma);
}

```

3. Imagine uma função que serve para somar dois valores como ilustrado na figura:

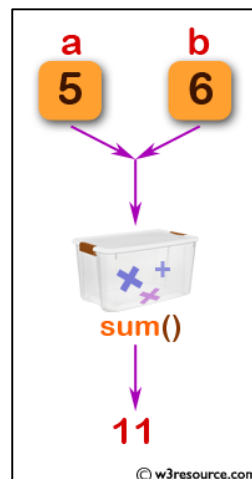


Figura A- 2 – Algoritmo

- a) Complete o seguinte cabeçalho da função, que contém os parâmetros.

```

int sum (*****){
    int s;
    s=a+b;
    return s;
}

```

- b) As variáveis s, a e b são globais ou locais?

4. O programa seguinte permite obter o preço de um produto ao final de 10 anos a uma certa taxa de inflação.

Além da função Main Existe a função “float produto_c_inflacao(float preco_atual)”, que calcula o preço inflacionado por ano.

```
#include <stdio.h>
#include <string.h>
float inflacao =3.5; //inflação anual de 3.5%
float preco=100; //preço atual do produto

float produto_c_inflacao(float preco_atual);

int main()
{
    int i=0;
    printf("a inflação anual é de: %f \n", inflacao);
    for (i=1;i<2;i++){
        preco=produto_c_inflacao(preco);
    }
    printf ("Ao fim de 10 anos, o preço final de um produto que custa hoje 100 €,
a uma inflação anual de %f, será de:%f €", inflacao, preco);
    return 0;
}
//-----//
float produto_c_inflacao(float preco_atual){
    float preco_final=preco_atual*(1+inflacao/100);//exemplo:
    preco_final=100*1.035
    return preco_final;
}
```

- a) Neste exemplo a função produto_c_inflacao(float preco_atual) permite obter o preço inflacionado por cada ano. Mas se pretendesse usar a mesma função para obter o preço inflacionado para n anos, seria possível ou útil usar recursividade nesta função?

- b) O que é recursividade de funções?

Anexo A1 – Correção do teste diagnóstico

Aluno /Pergunta	1a	1b	2a	2b	3a	3b	4a	4b	Nota Final
I		10							1
H	100	80					100	100	48
C		100			100			100	38
D	100	100	10	90		100	50		56
A		10							1
G						100		50	19
B					50				6
M	100	100	50			100		50	50
K	100	100	10			100		50	45
J		100							13
E	100	90							24
F		90					100	100	36
L		100							13

Pergunta	Cotação
1a	12,5
1b	12,5
2a	12,5
2b	12,5
3a	12,5
3b	12,5
4a	12,5
4b	12,5

Anexo B – Apresentação Expositiva de conceitos em PowerPoint



Intervenção pedagógica supervisionada

AGRUPAMENTO DE ESCOLAS EDUARDO GAGEIRO

Curso profissional de técnico de gestão e programação de sistemas informáticos

10.º ano - Programação e Sistemas de Informação
Modulo 4 - Estruturas de Dados Estáticas

Orientadores: Prof. Dr. João Piedade | Prof. Dr. Ana Paula Afonso
Cooperante: Prof. º Cláudia Barata
Aluno: Paulo de Oliveira Ferreira N.º 11840

Variáveis estruturadas - Strings



aula 1	03/03/2020	Conceitos exercícios
aula 2	05/03/2020	Exercícios Início atividade Ardumotic
aula 3	10/03/2020	Atividade Ardumotic



Variáveis Estruturadas - Arrays e Strings

► Variáveis simples

- `Int altura = 150;`
- `Float Media;`
- `Char='a';`

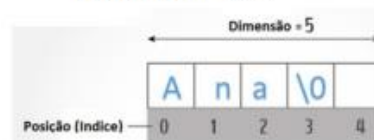
Variáveis estruturadas

► Array

`int drone_altitudes_recolhidas[1000];`

► String

`Char nome[5] = "Ana";`



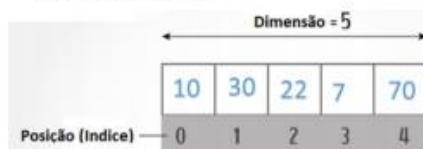
Nome[0] -> A
Nome[1] -> n
Nome[2] -> a
Nome[3] -> \0
Nome[4] -> ?

`Printf("o nome do aluno é: %s", nome);`
`Printf("a primeira letra do nome é : %c", Nome[0]);`

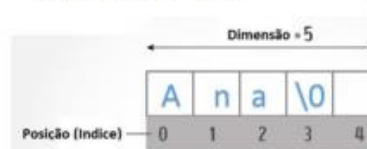
Execute o exercício 1

Definição de String?

`Int dimensoes[5];`



`Char nome[5] = "Ana"`

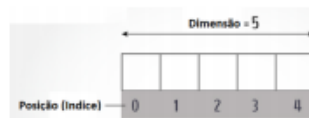


- String é uma variável estruturada na forma de Array do tipo char.
- Array vs string
 - a seguir ao último caracter existe o "\0" (elemento zero da tabela ASCII), e que é atribuído automaticamente quando a String é inicializada ou alterada.
- Observar tabela ASCII e porquê o '\0'?

Bin	Oct	Dec	Hex	Abrev	Notação com circunflexo	Código escape	Nome
0000 0000	000	00	00	NUL	^@	\0	Nulo (inglês Null)
0000 0001	001	01	01	SOH	^A		Início de cabeçalho (inglês Start of Header)
0000 0010	002	02	02	STX	^B		Início de texto (inglês Start of Text)

Formas de inicialização de Strings

```
Char nome[5];  
Nome = "Ana";
```



```
char Str4[] = "Arduino";
```



Executar exercício 2
Aprender a usar o depurador DEV C++

Diferença entre "A" e 'A'

```
char letra1='A'  
Char string1[]="A";
```

Endereço	Conteúdo	Variável
...
2999	A	letra1
3000		
3001	A	string1
3002	\0	
3003		
3004		
3005		

Strings e ponteiros para Strings

Char Str1[]="ABC";



Char Str1[]="ABC";
char* str2;



Char Str1[]="ABC";
char* str2;
str2=Str1;

Endereço	Conteúdo	Variável
...
2999	A	Str1
3000	B	
3001	C	
3002		
3003		
3004		
3005		

Endereço	Conteúdo	Variável
...
2999	A	Str1
3000	B	
3001	C	
3002		
3003	NULL	str2
3004		
3005		

Endereço	Conteúdo	Variável
...
2999	A	Str1
3000	B	
3001	C	
3002		
3003	2999	str2
3004		
3005		

printf("%s",str1); // output: ABC
printf("%s",str2); //output: ABC

Executar
exercício 3

Strings e ponteiros para Strings (cont.)

- Vantagens dos ponteiros para String
 - Podemos alterar o "conteúdo" simplesmente alterando o endereço para onde aponta.

Char str3[]="DEF";
Str2=str3;

Endereço	Conteúdo	Variável
...
2999	A	Str1
3000	B	
3001	C	
3002		
3003	3006	Str2
3004		
3005		
3006	D	Str3
3007	E	
3008	F	

Ponteiros para String e retorno de algumas funções

exercício 7

Exemplo de Funções:

Int strcmp (const char *, const char *)

Char * strcat (char *, const char *)

Quando se espera uma String no retorno, passar apenas o endereço dessa String, torna a execução mais rápida:

Código exemplo:

exercício 4 e 6

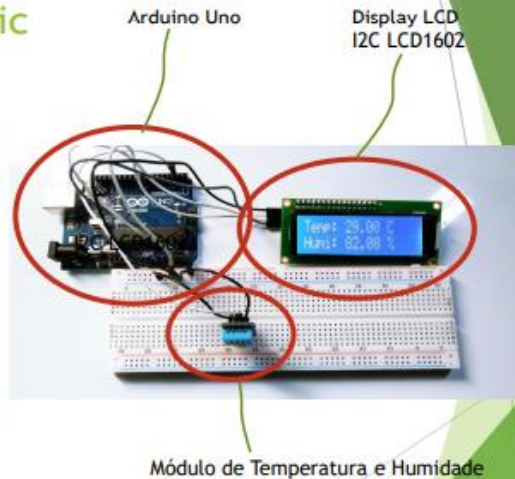
```
Char nome [100];
Char sobrenome [10];

Char nome_completo[100];
char* nome_completo_pointer;

Nome_completo=strcat (nome, sobrenome); //errado
Nome_completo_pointer=strcat (nome, sobrenome); //correto

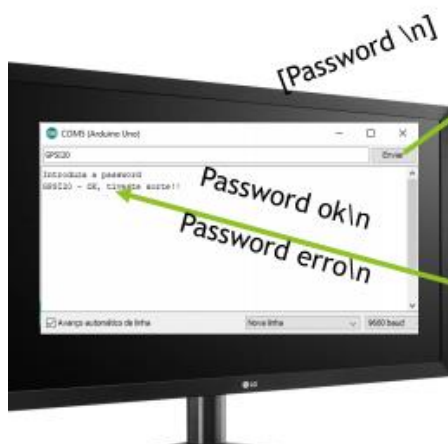
Printf("nome completo:%s", nome);
Printf("nome completo:%s", nome_completo_pointer);
```

Atividade Ardumotic



Módulo de Temperatura e Humidade

Envio de Strings Monitor série <-> Arduino



```
do {
    c=Serial.read();
    str_recebida[n]=c;
    n++;
    delay(2);
} while(c!='\n');

str_recebida[n-1]='\0';
Serial.print(str_recebida);

if (strcmp(str_recebida, str_password)==0)
{
    Serial.println(" - OK");
}
else
{
    Serial.println(" - Erro");
}
```



Anexo C – Ficha de exercícios em DEV C++

Os seguintes exercícios contêm código completo ou parcial de programas, a testar no IDE DEV C++.

Algumas respostas devem ser dadas textualmente nesta ficha de exercícios, outras que envolvem código a implementar no DEV c++. O script final de cada exercício deve guardado na sua pasta web habitual com identificação clara do exercício correspondente.

Exercicio 1

A aprender:

- Atribuição de valor a uma String e visualização do seu conteúdo.

O seguinte programa permite introduzir um nome e devolver a respetiva primeira letra com visualização no ecrã.

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char** argv)
{
    char nome[10];
    printf("Escreva o seu primeiro nome: ");
    scanf("%s", nome);
    printf("Ok, tu chamas-te: %s\n", nome);
    printf("A primeira letra do teu nome é:%c", nome[0]);
    return 0;
}
```

Observar:

O “%s” é o operador para representar uma variável String, neste caso a variável *nome*. Mas quando queremos obter apenas um dos caracteres da String usamos o operador %c.

Questões:

- a) Tal como está a ser feito para apresentar a primeira letra do nome, use um ciclo “for” para mostrar todas as letras da String nome (os 10 caracteres do array).

Resolução:

```

#include <stdio.h>
#include <stdlib.h>
int main(int argc, char** argv)
{
    char nome[10];
    printf("Escreva o seu primeiro nome: ");
    scanf("%s", nome);
    printf("Ok, tu chamas-te: %s\n", nome);
    printf("A primeira letra do teu nome é:%c", nome[0]);
    for(int i=0; i<10; i++){
        printf("A letra %i do teu nome é:%c", i, nome[i]);
    }
    return 0;
}

```

- b) O que acontece quando introduz no teclado uma String com a dimensão máxima (10 caracteres) ou superior à da variável str?

Exercicio 2

A aprender:

- Calcular a dimensão de uma String introduzida pelo teclado.
- Diferença entre fgets e scanf

O seguinte programa permite criar uma String e introduzir o seu conteúdo pelo teclado. A seguir apresenta a dimensão da String introduzida.

```

#include <stdio.h>
#include <stdlib.h>
int main(int argc, char** argv)
{
    char str[10]; /* Declara uma String de dimensão 10 */
    int l= 0;
    printf("\n\nCalcular o comprimento de uma String :\n");
    printf("Introduza uma string : ");
    fgets(str, sizeof str, stdin);
    while(str[l]!='\0')
    {
        l++;
    }
    printf("O comprimento da String introduzida é: %d\n", l-1);
    return 0;
}

```

Questões:

- a) Use o depurador do DEV C++ para observar todos os caracteres que efetivamente ficaram guardados na String *str*, colocando um breakpoint na última linha;

Consegue justificar porque se colocou “l-1” e não apenas “l” na linha:

```
printf("O comprimento da String introduzida é: %d\n", l-1);
```

Resposta:

O método `fgets` coloca um ‘\n’ no fim da string, enquanto o `scanf` recebe apenas a primeira palavra e sem ‘\n’

- b) Se em vez do método:

```
fgets(str, sizeof str, stdin);
```

Utilizasse o método:

```
scanf("%s",str);
```

Qual a diferença em termos de conteúdo da String? Visualize com atenção o output em ambos os casos, e utiliza o depurador como auxílio de interpretação, tal como na alínea anterior.

Resposta:

o método `fgets` coloca um ‘\n’ no final da String, enquanto o `scanf` recebe apenas a primeira palavra e sem ‘\n’.

- c) A função `STRLEN` da API permite obter diretamente o comprimento de uma String. Adapte o programa usando a função `STRLEN`.
Exemplo:

```
Char string1[]="gpsi";  
Int comprimentoastring=strlen(string1);
```

Resposta:

```
printf("O comprimento da String introduzida é: %d\n", strlen(str)-1);
```

Exercicio 3

A aprender:

- String vs ponteiros para String

Observe e teste o seguinte programa:

```
#include <stdio.h>  
#include <string.h>  
int main()  
{  
    char str1[] = "Gestão e programação de sistemas informáticos";  
    char str2[] = "Programação de Sistemas Informáticos";
```

```

char* str_pointer1=str1;
char* str_pointer2=str2;

printf("Endereço da string1: %X\n",str1);
printf("Endereço da string2: %X\n",str2);
printf("Endereço de str_pointer1: %X\n",str_pointer1);
printf("conteudo da localizacao apontada por str_pointer1:
%s\n",str_pointer1);

str_pointer1=str2;
printf("Novo endereço de str_pointer1 é: %X\n",str_pointer1);
printf("Novo conteudo da localizacao apontada por str_pointer1:
%s\n",str_pointer1);
return 0;
}

```

Observações:

Para visualizarmos o endereço, quer seja de um ponteiro para *string* (*char* xpto*) ou uma *String* explicita *char[] xpto*, usa-se o operador *%X* (onde *X* significa hexadecimal, que é o formato dos endereços de memória).

Questões:

- Coloque as duas linhas de código seguinte no programa e execute. Explique o que acontece.

```
str1=str2;
```

Resposta:

As strings não permitem atribuição direta do seu conteúdo como “*str1=str2*”, em substituição deve usar-se a função *strcpy*:

```
Strcpy(str1,str2);
```

```
str_pointer1=str2;
```

Resposta: o endereço de *str_pointer1* passa a ser o da string *str2*.

- Qual a diferença entre uma *String* e um ponteiro para *String*?

Exercicio 4

A aprender:

- Concatenação de Strings.

A função da linguagem C que permite efetuar concatenações é a função *STRCAT*. Esta função Acrescenta a *String src* ao final da *String dest*. Deve certificar-se que a *String dest* tem capacidade para as duas.

A sua estrutura e parâmetros são:

*char *strcat(char *dest, const char *src)*

- *dest*: String de destino;
- *src*: String a copiar para *dest*, acrescentando ao seu conteúdo a partir do último caracter.

Retorno: ponteiro para uma String;

Agora observa o seguinte programa que pede o primeiro e ultimo nome separadamente em três formas de o fazer:

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char** argv)
{
    char nome[100];
    char sobrenome[10];

    printf("Escreva o seu primeiro nome: ");
    scanf("%s", nome);
    printf("Escreva o sobrenome: ");
    scanf("%s", sobrenome);
    strcat(nome, sobrenome);
    printf("o nome completo é : \n %s", nome);
    return 0;
}
```

Script 1 - Concatenação versão C

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char** argv)
{
    char nome[100];
    char sobrenome[10];

    printf("Escreva o seu primeiro nome: ");
    scanf("%s", nome);
    printf("Escreva o sobrenome: ");
    scanf("%s", sobrenome);

    strcat(nome, " "); //concatenação do
    //carater espaço à string nome
    strcat(nome, sobrenome); //concatenação
    //da string sobrenome à string nome

    printf("o nome completo\n");
    printf("%s\n", nome);

    return 0;
}
```

Script . 3 – Concatenação versão A

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char** argv)
{
    char nome[10];
    char sobrenome[10];
    char nome_completo[100];

    printf("Escreva o seu primeiro nome: ");
    scanf("%s", nome);
    printf("Escreva o sobrenome: ");
    scanf("%s", sobrenome);

    strcat(nome_completo, nome); //concatenação
    //da string nome à string nome_completo
    strcat(nome_completo, " "); //concatenação
    //do carater espaço à string nome_completo

    strcat(nome_completo, sobrenome); //concatenação
    //da string sobrenome à string
    //nome_completo
}
```

Script . 2 - Concatenação versão B

Questões:

- c) Executa a versão A. Nesta versão deves ter observado que ao fazer a concatenação do primeiro nome com o sobrenome o resultado foi uma String sem espaço de separação entre os dois.
- a) Para corrigir este “problema” temos as versões B e C.

Na versão B: fazemos concatenações sucessivas à String “nome”:

1. Primeira concatenação:
Adicionar um caracter “espaço” à String nome;
2. Segunda concatenação:
adicionar a String sobrenome à String nome;

Na versão C: é utilizada uma terceira String “nome_completo”, de modo a fazer concatenações sucessivas para esta String até ter o nome completo, sem alterar a String nome ou sobrenome:

1. Primeira concatenação:

- ```
nome_completo->nome_completo+ nome;
```
2. Segunda concatenação:

```
nome_completo->nome_completo+" ";
```
  3. Terceira concatenação:

```
nome_completo->nome_completo+sobrenome;
```

Substitui os “??” nessas duas versões b e c de modo a ter o output final de nome completo corretamente.

## Exercicio 5

### A aprender:

- Inversão de Strings

Exemplo:

Receber uma String do teclado e devolver essa String invertida:

“gato” ->”otag”

Código completo:

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char** argv)
{
 char str[100]; /* Declara uma string de dimensão 100 */
 int l,i;
 printf("Introduza uma string : ");
 //fgets(str, sizeof str, stdin); //deve ser substituído pela linha seguinte
 scanf("%s",str);
 l=strlen(str);
 printf("os caracteres que introduziu, na ordem inversa são : \n");
 for(i=l;i>=0;i--)
 {
 printf("%c ", str[i]);
 }
 printf("\n");
 return 0;
}
```

### Observações:

- O ciclo “for” usa o decremento da variável j em cada iteração. Neste caso convém, porque pretendemos começar pelo último carácter de modo a enviar para o ecrã em ordem inversa.

### Questões:

- d) Se a String original for igual à String invertida, e tratando-se de uma palavra apenas, então essa palavra é o que se chama de palíndromo. Acrescente o

código necessário para confirmar ao utilizador se a palavra introduzida é um palíndromo.

Sugestão:

Criar uma segunda String à qual seja atribuído o conteúdo da primeira invertido.

- e) A função STRREV permite obter diretamente uma String invertida.

Implemente a alteração necessária para utilizar essa função.

Grava o script final com a identificação do exercício.

### Resolução:

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char** argv)
{
 char str[100]; /* Declara uma string de dimensão 100 */
 char str_clone[100];

 int l,i;
 printf("Introduza uma string : ");
 scanf("%s",str);
 l=strlen(str);
 printf("os caracteres que introduziu, na ordem inversa são : \n");
 for(i=l;i>=0;i--)
 {
 printf("%c ", str[i]);

 }
 printf("\n");
 strcpy(str_clone,str);
 strrev(str);
 if (strcmp(str,str_clone)==0){
 printf("a palavra introduzida é um palindrome");
 }
 else{
 printf("a palavra introduzida nao é um palindrome");
 }
 return 0;
}
```

## Exercicio 6

### A aprender:

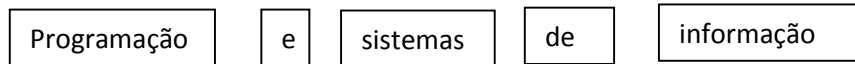
Split (divisão) de uma String em varias strings - função STRTOK

### Exemplo:

Dividir uma String pelo carater “ ” (espaço)

String: “Programação e sistemas de Informação”

Resultado: 5 Strings isoladas



Chama-se token a cada palavra ou conjunto de palavras separadas pelo delimitador (parâmetro delim) e daí o nome da função STR (String) + TOK(de token).

A função tem a seguinte estrutura:

*Char \* strtok(char\* s, const char\* delim)*

Parâmetros:

- Char\* s – String principal, na qual queremos efetuar o split;
- Const char\* delim – caracter ou palavra que separa cada token.

Implementação:

O primeiro token é obtido normalmente da seguinte forma:

*char\* token = strtok(str, " ")*

Nos tokens seguintes a diferença é que devemos substituir a variável str por NULL, é assim que o compilador funciona para saber que estamos a continuar a pedir tokens da mesma String.

E como não sabemos à partida quantos tokens haverá, devemos utilizar um ciclo while:

```
while (token != NULL) {
 printf("%s\n", token);
 token = strtok(NULL, " ");
}
```

**Código completo:**

```
#include <stdio.h>
#include <string.h>

int main()
{
 char str[] = "Gestao e programação de sistemas informaticos";
 // Obter o primeiro token
```

```

char* token = strtok(str, " ");
// obter os token seguinte(se houver mais, ou seja enquanto for
diferente de NULL)
while (token != NULL) {
 printf("%s\n", token);
 token = strtok(NULL, " ");
}
return 0;
}

```

### Questões:

- Implemente um programa idêntico para a String seguinte, sendo o carater delimitador o “#”:  
“hugo:15#joana:12#vasco:18”
- Acrescente o código necessário para contar o número de tokens encontrados e apresenta-los no output.

### Resolução:

```

#include <stdio.h>
#include <string.h>
int main()
{
 int num_tokens=0;
 char str[] = "hugo:15#joana:12#vasco:18";
 // Obter o primeiro token
 char* token = strtok(str, "#");
 // obter os token seguinte(se houver mais, ou seja enquanto for
diferente de NULL)
 while (token != NULL) {
 printf("%s\n", token);
 token = strtok(NULL, "#");
 num_tokens++;
 }
 printf("foram encontrados %d tokens", num_tokens);
 return 0;
}

```

### Exercicio 7

Função de comparação STRCMP.

Esta função permite comparar duas Strings permitindo-nos saber se são iguais ou não.

Estrutura da função:

```
Int strcmp(const char* s1, const char* s2)
```

Parâmetros:

S1 é a string1

S2 é a string2

Se S1 for exatamente igual a s2 então esta função retorna 0.

Exemplo:

```
#include<stdio.h>
#include<string.h>
int main()
{
 char string1[] = "felino";
 char string2[] = "felino";
 int compara = strcmp(string1, string2);
 if (compara==0)
 printf("as duas strings são iguais");
 else
 printf("as duas strings são diferentes");
 return 0;
}
```

### Questões:

- a) Imagine que este programa podia ser utilizado para verificar uma palavra-chave, onde string1 seria uma password introduzida pelo utilizador através do scanf, e a string2 seria a password esperada.

Alterações necessárias

- O valor da string1 é recebido pelo scanf;
- Consoante o resultado da comparação das duas Strings, alterar a mensagem para “Login ok” ou “Password errada” respetivamente;

Resolução:

```
#include<stdio.h>
#include<string.h>
int main()
{
 char string1[] = "felino";
 char string2[20]; //presumindo que a palavra chave nao tem mais que
19 caracteres
 printf("introduza a palavra chave:\n");
 scanf ("%s",string2);
 int compara = strcmp(string1, string2);
 if (compara==0)
 printf("login ok");
 else
 printf("palavra chave errada!!");
 return 0;
}
```

### Errata

- No exercício 1, no ponto “Observar” , foi feita a correção de substituir “variável str” por “variável nome”;
- No exercício 5 a), no script apresentado como exemplo, na linha 13:

A expressão:

*for(i=l;i>=0;i--)*

Deve corrigir-se para:

*for(i=l-1;i>=0;i--)*

## Anexo D - Teste de Aferição

Através das seguintes questões pretende-se avaliar de forma resumida os conhecimentos básicos sobre as variáveis String no enquadramento das Estruturas de Dados Estáticas da Programação em C.

**Nas questões seguintes escolha a opção correta:**

1. Nas variáveis String, o símbolo “\0” é um:
  - Símbolo da tabela ASCII para fazer quebras de linha;
  - Símbolo da tabela ASCII para identificar o final de uma String;
  - Uma divisão por zero.

2. Observe:

```
char nome[]="oscarneiro"
printf("%c", nome[3]);
```

Escolha a opção que corresponde ao output correto:

- c
- a
- r

3. Observe:

```
char nome[10]="Omar";
char sobrenome[10]="Melada";
char* str_pointer; //ponteiro para string
```

Qual das seguintes expressões irá dar um erro?

- sr\_pointer=nome;
- sobrenome="Melo";

4. A função strcmp serve para:
  - Comprimir Strings;
  - Comparar Strings;
  - Obter o comprimento de uma String.
5. Observe com muita atenção:

```
char string1[15]="Manuel";
char string2[15]="Santos";
strcat(string2,string1);
Printf("%s", string1);
```

Escolha o output correto:

- Manuel
- ManuelSantos
- santosManuel

6. A função strtok serve para:
  - Verificar se a String esta bem construída;
  - Dividir uma String em partes.

## Anexo E1 – Planificações das aulas do dia 3/03/20

| Plano de Aulas – dia 3/03/20                                                                                                                                                                                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                                                                                                                |              |                                                                                                                                                                                                                                                                                                    |                                                                                                                                                     |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| Escola Secundária de Sacavém - Curso profissional de técnico de gestão e programação de sistemas informáticos                                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                                                                                                                |              |                                                                                                                                                                                                                                                                                                    |                                                                                                                                                     |
| Disciplina:                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Programação e Sistemas de Informação                                                                                                           |              | 10.º Ano                                                                                                                                                                                                                                                                                           |                                                                                                                                                     |
|                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                                                                                                                |              | Ano Letivo: 2019/2020                                                                                                                                                                                                                                                                              |                                                                                                                                                     |
| 2º Período                                                                                                                                                                                                                                 | Modulo 4 - Estruturas de Dados Estáticas                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Data:                                                                                                                                          |              | Hora:                                                                                                                                                                                                                                                                                              |                                                                                                                                                     |
|                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Aula nº: 1,2 e 3                                                                                                                               |              | Duração: 3 x 45 Min.                                                                                                                                                                                                                                                                               |                                                                                                                                                     |
|                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Sala: C218                                                                                                                                     |              | Professor: Paulo Ferreira                                                                                                                                                                                                                                                                          |                                                                                                                                                     |
| Objetivos principais:                                                                                                                                                                                                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                                                                                                                |              |                                                                                                                                                                                                                                                                                                    |                                                                                                                                                     |
| <ul style="list-style-type: none"><li>▪ Distinção entre uma variável simples e uma variável estruturada.</li><li>▪ Conhecer o conceito de String.</li><li>▪ Manipular uma String através das funções da biblioteca C do DEV C++.</li></ul> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                                                                                                                |              |                                                                                                                                                                                                                                                                                                    |                                                                                                                                                     |
| Conteúdos                                                                                                                                                                                                                                  | Objetivos de aprendizagem                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Estratégias/Atividades                                                                                                                         | Tempo (Min.) | Recursos educativos                                                                                                                                                                                                                                                                                | Avaliação                                                                                                                                           |
| <ul style="list-style-type: none"><li>•String como um Array de caracteres;</li><li>•Estrutura de uma String e respetivas formas de manipulação.</li><li>•Tabela ASCII</li><li>•Biblioteca de funções de manipulação de Strings</li></ul>   | <ul style="list-style-type: none"><li>• Distinguir entre uma variável simples e uma variável estruturada.<ul style="list-style-type: none"><li>•Entender a diferença e semelhança entre uma String e um Array de caracteres;</li><li>•Entender a função do símbolo “\0” e a sua representação na tabela ASCII.</li></ul></li><li>• Conhecer a estrutura de armazenamento da String em memória.</li><li>• Constatar a necessidade de utilização de dados estruturados em Strings, e conhecer as funções disponibilizadas para a sua manipulação na linguagem C;</li></ul> | <ul style="list-style-type: none"><li>• Apresentação aos alunos.</li><li>• Propor a execução de uma breve ficha de diagnóstico.</li></ul>      | 45 Min.      | <ul style="list-style-type: none"><li>• Projetor de vídeo;</li><li>• Apresentação em PPT relativa aos conteúdos a abordar;</li><li>• 1 PC por aluno com acesso à Internet;</li><li>• Grelha de observação de atitudes, valores e aptidões;</li><li>• Documentos impressos por aluno com:</li></ul> | <ul style="list-style-type: none"><li>• Conhecimento e capacidades cognitivas 50%</li><li>• Atitudes e valores 35%</li><li>• Aptidões 15%</li></ul> |
|                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | <ul style="list-style-type: none"><li>• Apresentação de diapositivos sobre os conceitos básicos e interação formativa com os alunos.</li></ul> | 15 Min.      |                                                                                                                                                                                                                                                                                                    |                                                                                                                                                     |
|                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | <ul style="list-style-type: none"><li>• Executar os exercícios 1 até ao 4 acompanhando dentro do possível com a</li></ul>                      | 75 Min.      |                                                                                                                                                                                                                                                                                                    |                                                                                                                                                     |



Planificações das aulas do dia 3/03/20 - continuação

|                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                                    |                                 |  |                                                                                                                                                                                                           |  |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|--|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
|                                                                                                                                                                                            | <ul style="list-style-type: none"> <li>• Executar exercícios tipo problema, em forma simplificada, de: <ul style="list-style-type: none"> <li>• Criação e visualização de Strings</li> <li>• Obter dimensão de Strings e dominar o acesso a todos os caracteres da String, incluindo o '\0'</li> <li>• Distinguir entre Strings e ponteiros para Strings</li> <li>• Concatenação de Strings</li> </ul> </li> </ul> | apresentação expositiva do PPT. |  | <ul style="list-style-type: none"> <li>○ Manual teórico com lista de funções de manipulação de Strings</li> <li>○ Tabela ASCII.</li> <li>○ Ficha de exercícios disponível também na web Cloud;</li> </ul> |  |
| <p>■ Fontes: Direção-Geral Formação Vocacional. Programa do Curso profissional de técnico de gestão e programação de sistemas informáticos. Lisboa: Ministério da Educação/DGFV, 2005.</p> |                                                                                                                                                                                                                                                                                                                                                                                                                    |                                 |  |                                                                                                                                                                                                           |  |

## Anexo E2 – Planificações das aulas do dia 05/03/20

[illegible]

Planificações das aulas do dia 05/03/20 - continuação

|                                                                                                                                                                                            |                                                                                                                                                                                                                                                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |  |                                                                                                                                                                                                                                                                                                                                                                                                           |  |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
|                                                                                                                                                                                            | <ul style="list-style-type: none"> <li>• Executar com sucesso os scripts de teste: <ul style="list-style-type: none"> <li>○ Do módulo DHT11 com a comunicação com o Monitor-Serie,</li> <li>○ Do display LCD i2c com as funções principais da sua API;</li> </ul> </li> </ul> | <ul style="list-style-type: none"> <li>• Efetuar as ligações elétricas dos módulos DHT11 e LCD I2C;</li> <li>• Upload para a IDE Arduino das bibliotecas fornecidas, indicando a localização na pasta de recursos;</li> <li>• Teste das funções de leitura dos sensores de temperatura e humidade, utilizando para isso o Monitor Série para receber informação;</li> <li>• Da mesma forma deverá orientar-se os alunos na realização dos testes ao display LCD com o script fornecido, com especial atenção para a perceção da sua estrutura e funções utilizadas.</li> </ul> |  | <ul style="list-style-type: none"> <li>○ Ficha de exercícios disponível também na web Cloud;</li> <li>• O seguinte material por grupo: <ul style="list-style-type: none"> <li>○ Um Arduino Uno com cabo USB;</li> <li>○ Display LCD 16×2 (IIC I2C TWI Serial 1602 16×2);</li> <li>○ 7 Fios conectores MF;</li> <li>○ Módulo com sensores de temperatura e humidade - Shield DHT-11</li> </ul> </li> </ul> |  |
| <p>■ Fontes: Direção-Geral Formação Vocacional. Programa do Curso profissional de técnico de gestão e programação de sistemas informáticos. Lisboa: Ministério da Educação/DGFV, 2005.</p> |                                                                                                                                                                                                                                                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |  |                                                                                                                                                                                                                                                                                                                                                                                                           |  |

## Anexo E3 – Planificações das aulas do dia 10/03/20

| Plano de Aulas – dia 10 de março                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                            |         |                                                                                                                                                                                                                                                                                                                     |                                                                                                                                                     |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| Escola Secundária de Sacavém - Curso profissional de técnico de gestão e programação de sistemas informáticos                                                                                                                                                                                                                                                                                                                                                                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                            |         |                                                                                                                                                                                                                                                                                                                     |                                                                                                                                                     |
| Disciplina: Programação e Sistemas de Informação                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                       | 10.º Ano                                                                                                                                                                                                                                                                                                                                                                                                                                                   |         | Ano Letivo: 2019/2020                                                                                                                                                                                                                                                                                               |                                                                                                                                                     |
| 2º Período                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Modulo 4 - Estruturas de Dados Estáticas                                                                                                                                                                                                                                                                                                                                                                                              | Data:                                                                                                                                                                                                                                                                                                                                                                                                                                                      |         | Hora:                                                                                                                                                                                                                                                                                                               |                                                                                                                                                     |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                       | Aula nº: 8,9 e 10                                                                                                                                                                                                                                                                                                                                                                                                                                          |         | Duração: 3 x 45m                                                                                                                                                                                                                                                                                                    |                                                                                                                                                     |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                       | Sala: C218                                                                                                                                                                                                                                                                                                                                                                                                                                                 |         | Professor: Paulo Ferreira                                                                                                                                                                                                                                                                                           |                                                                                                                                                     |
| Objetivos específicos:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                            |         |                                                                                                                                                                                                                                                                                                                     |                                                                                                                                                     |
| <ul style="list-style-type: none"><li>■ <b>Consolidação de conceitos e da definição de String - identificação que características específicas na edição de código na linguagem C, que a distinguem de uma variável simples;</b></li><li>■ <b>Manipular uma String.</b></li><li>■ <b>Reconhecer as formas principais de inicialização e manipulação de Strings</b></li><li>■ Cópia e atribuição de valor; cópia de subtrings; concatenações sucessivas da mesma String para obtenção de uma mais complexa;</li></ul> |                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                            |         |                                                                                                                                                                                                                                                                                                                     |                                                                                                                                                     |
| Conteúdos                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Objetivos de aprendizagem                                                                                                                                                                                                                                                                                                                                                                                                             | Estratégias/Atividades                                                                                                                                                                                                                                                                                                                                                                                                                                     | Tempo   | Recursos educativos                                                                                                                                                                                                                                                                                                 | Avaliação                                                                                                                                           |
| <ul style="list-style-type: none"><li>• <i>Estrutura de uma String e respetivas funções de manipulação:</i></li><li>• <i>Inicialização/atribuição de valor; cópia, concatenação e comparação.</i></li><li>• <i>Envio de Strings para o LCD do Arduino</i></li><li>• <i>Envio e receção de String para e do Monitor Série Arduino</i></li></ul>                                                                                                                                                                      | <ul style="list-style-type: none"><li>• Saber fazer a distinção entre uma variável simples e uma variável estruturada.</li><li>• Saber identificar a necessidade de utilização de dados estruturados em Strings, e dominar as funções disponibilizadas para a sua manipulação na linguagem C;</li><li>• Ser capaz de entender algoritmos para efetuar a transmissão e tratamento de Strings na atividade prática Ardumotic.</li></ul> | <ul style="list-style-type: none"><li>• Revisão de conceitos e correção dos principais exercícios anteriores;</li><li>• Início da realização da atividade Ardumotic.<ul style="list-style-type: none"><li>• Montagem;</li><li>• Visualização e execução de programas demonstrativos;</li></ul></li><li>• Resolução de questões para aplicar os conhecimentos adquiridos nesses programas;</li><li>• Preenchimento de um inquérito investigativo.</li></ul> | 135 Min | <ul style="list-style-type: none"><li>•Projeção de uma apresentação PPT, a partir do slide 10;</li><li>•Grelha de avaliação de atitudes e valores;</li><li>•Pasta na cloud para guarda trabalhos dos alunos;</li><li>•Exercícios sobre manipulação de Strings, disponibilizados aos alunos via Web Cloud.</li></ul> | <ul style="list-style-type: none"><li>• Conhecimento e capacidades cognitivas 50%</li><li>• Atitudes e valores 35%</li><li>• Aptidões 15%</li></ul> |
| <ul style="list-style-type: none"><li>■ Fontes: Direção-Geral Formação Vocacional. Programa do Curso profissional de técnico de gestão e programação de sistemas informáticos. Lisboa: Ministério da Educação/DGFV, 2005.</li></ul>                                                                                                                                                                                                                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                            |         |                                                                                                                                                                                                                                                                                                                     |                                                                                                                                                     |

## Anexo E4 – Planificações das aulas do dia 12/03/20

| Plano de Aulas – dia 12 de março                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |             |                                                                                                                                                                                                                                                                                                                                 |                                                                                                                                               |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| Escola Secundária de Sacavém - Curso profissional de técnico de gestão e programação de sistemas informáticos                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |             |                                                                                                                                                                                                                                                                                                                                 |                                                                                                                                               |
| Disciplina:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Programação e Sistemas de Informação                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 10.º Ano    |                                                                                                                                                                                                                                                                                                                                 | Ano Letivo: 2019/2020                                                                                                                         |
| 2º Período                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Modulo 4 - Estruturas de Dados Estáticas                                                                                                                                                                                                                                                                                                                                                                                         | Data:                                                                                                                                                                                                                                                                                                                                                                                                                                           |             | Hora:                                                                                                                                                                                                                                                                                                                           |                                                                                                                                               |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                                                                                                                                                                  | Aula nº:                                                                                                                                                                                                                                                                                                                                                                                                                                        | 11,12,13,14 | Duração:                                                                                                                                                                                                                                                                                                                        | 4 x 45m                                                                                                                                       |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                                                                                                                                                                  | Sala:                                                                                                                                                                                                                                                                                                                                                                                                                                           | C218        | Professor:                                                                                                                                                                                                                                                                                                                      | Paulo Ferreira                                                                                                                                |
| Objetivos específicos:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |             |                                                                                                                                                                                                                                                                                                                                 |                                                                                                                                               |
| <ul style="list-style-type: none"><li>Consolidação de conceitos e da definição de String - identificação que características específicas na edição de código na linguagem C, que a distinguem de uma variável simples;</li><li>Manipular uma String.</li><li>Reconhecer as formas principais de inicialização e manipulação de Strings<ul style="list-style-type: none"><li>Cópia e atribuição de valor; cópia de substrings; concatenações sucessivas da mesma String para obtenção de uma mais complexa;</li></ul></li></ul> |                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |             |                                                                                                                                                                                                                                                                                                                                 |                                                                                                                                               |
| Conteúdos                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Objetivos de aprendizagem                                                                                                                                                                                                                                                                                                                                                                                                        | Estratégias/Atividades                                                                                                                                                                                                                                                                                                                                                                                                                          | Tempo       | Recursos educativos                                                                                                                                                                                                                                                                                                             | Avaliação                                                                                                                                     |
| <ul style="list-style-type: none"><li>Estrutura de uma String e respetivas funções de manipulação:</li><li>Inicialização/atribuição de valor; cópia, concatenação e comparação.</li><li>Envio de Strings para o LCD do Arduino</li><li>Envio e receção de String para e do Monitor Série Arduino</li></ul>                                                                                                                                                                                                                     | <ul style="list-style-type: none"><li>Saber fazer a distinção entre uma variável simples e uma variável estruturada.</li><li>Saber identificar a necessidade de utilização de dados estruturados em Strings, e dominar as funções disponibilizadas para a sua manipulação na linguagem C;</li><li>Ser capaz de entender algoritmos para efetuar a transmissão e tratamento de Strings na atividade prática Arduomatic.</li></ul> | <ul style="list-style-type: none"><li>Revisão de conceitos e correção dos principais exercícios anteriores;</li><li>Início da realização da atividade Arduomatic.<ul style="list-style-type: none"><li>Montagem;</li><li>Visualização e execução de programas demonstrativos;</li></ul></li><li>Resolução de questões para aplicar os conhecimentos adquiridos nesses programas;</li><li>Preenchimento de um inquérito investigativo.</li></ul> | 190 Min     | <ul style="list-style-type: none"><li>Projeção de uma apresentação PPT, a partir do slide 10;</li><li>Grelha de avaliação de atitudes e valores;</li><li>Pasta na <a href="#">cloud</a> para guarda trabalhos dos alunos;</li><li>Exercícios sobre manipulação de Strings, disponibilizados aos alunos via Web Cloud.</li></ul> | <ul style="list-style-type: none"><li>Conhecimento e capacidades cognitivas 50%</li><li>Atitudes e valores 35%</li><li>Aptidões 15%</li></ul> |
| <ul style="list-style-type: none"><li>Fontes: Direção-Geral Formação Vocacional. Programa do Curso profissional de técnico de gestão e programação de sistemas informáticos. Lisboa: Ministério da Educação/DGFV, 2005.</li></ul>                                                                                                                                                                                                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |             |                                                                                                                                                                                                                                                                                                                                 |                                                                                                                                               |

## Anexo F - Respostas aos questionários investigativos

### Respostas agrupadas dos questionários aos alunos

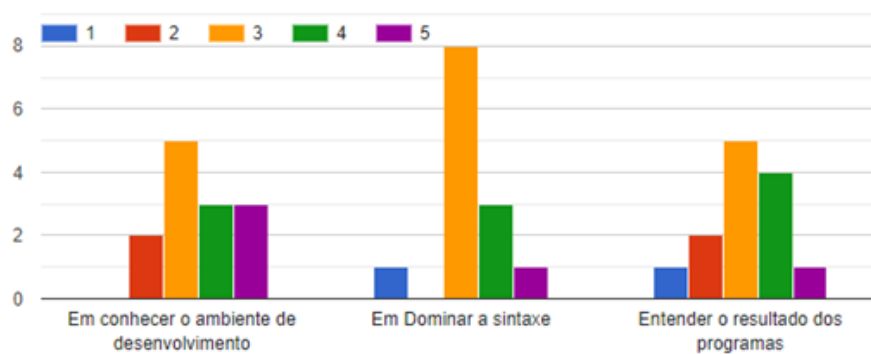
Os gráficos que se seguem contêm as respostas às respetivas questões colocadas nos questionários do final da primeira e segunda parte da intervenção respetivamente, agrupando a resposta do primeiro com a equivalente do segundo.

A primeira e segunda parte da intervenção são respetivamente designadas por DEV C++ e Arduino.

#### 1. Grau de dificuldade no Interface de Desenvolvimento

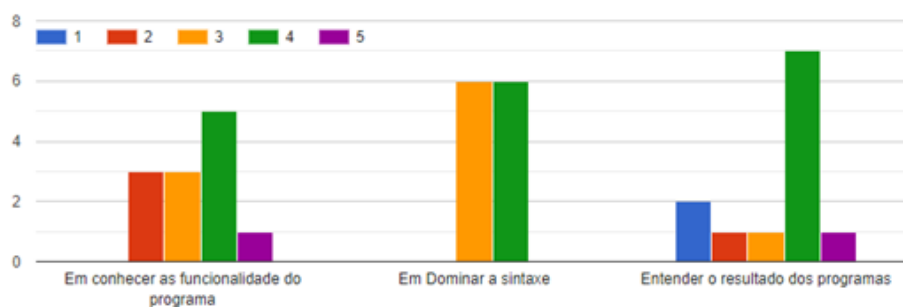
##### a) NO DEV C++

Questão: “No DEV C++ qual o grau de dificuldade? (1-pouca;5-muita)”



##### b) No Arduino

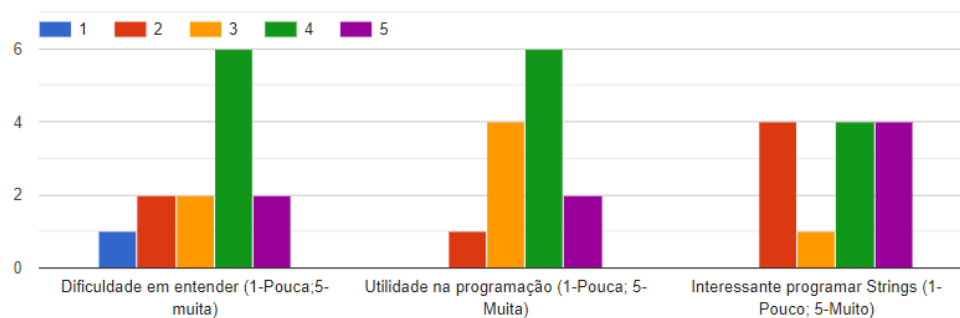
Questão: “Qual o grau de dificuldade que sentiu no ambiente de trabalho Arduino? (1-pouca; 5-muita) ”



#### 2. Sobre a temática Strings:

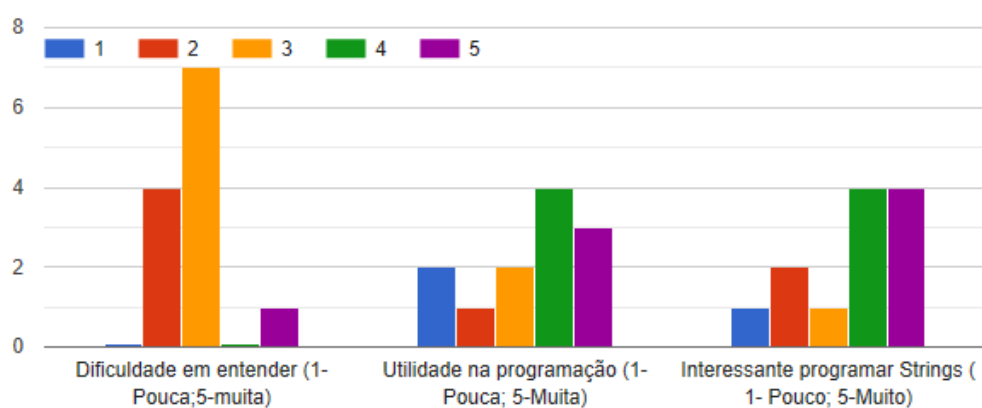
##### c) No DEV C++

Questão: “Nesta temática Strings considero:”



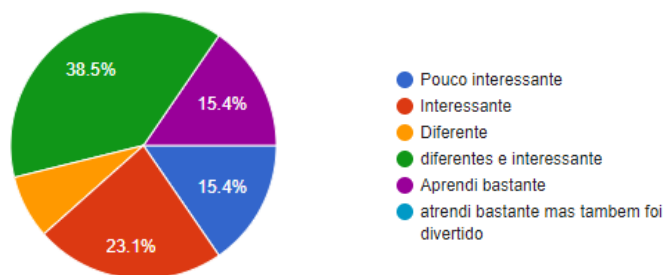
#### d) No Arduino

Questão: “Nesta temática Strings considero:”

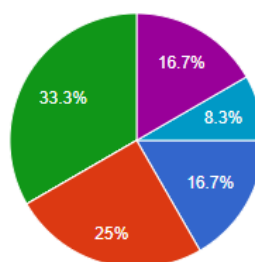


### 3. As aulas desta intervenção foram:

#### No DEV C++:

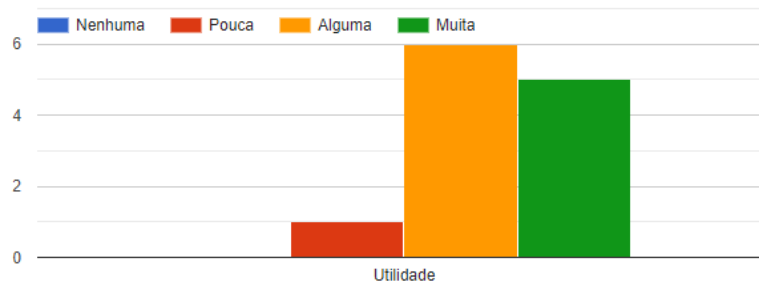


#### No Arduino:

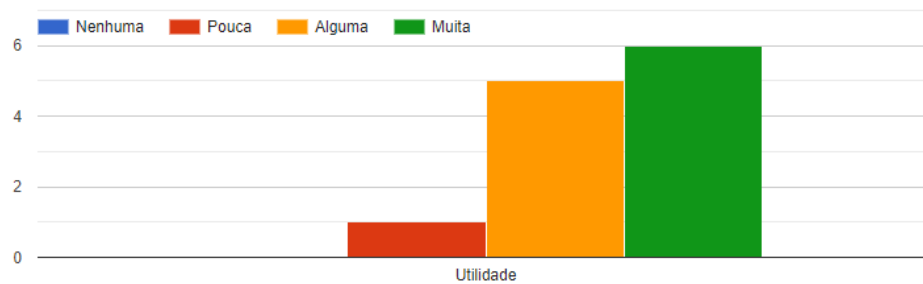


**4. Qual o grau de utilidade que acha que as Strings podem ter na programação de coisas reais?**

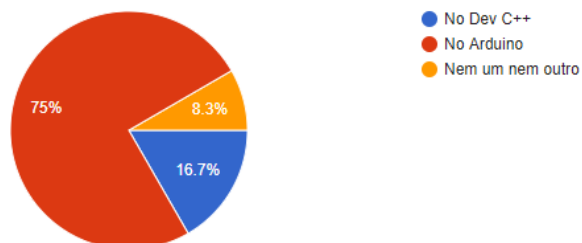
e) No DEV C++



f) No Arduino



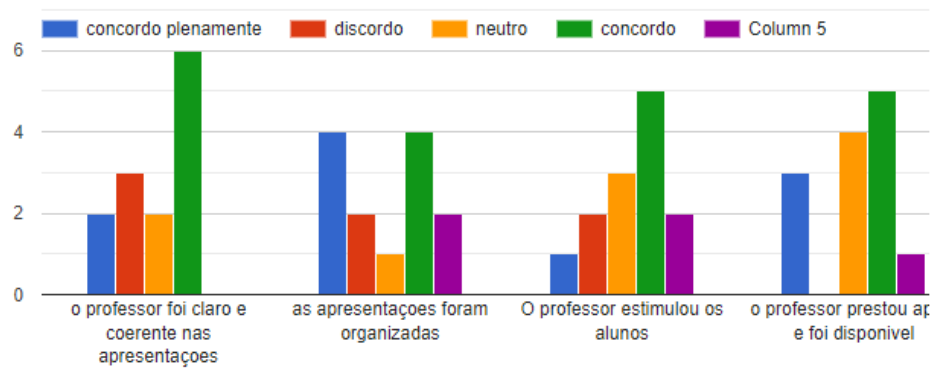
**5. É mais interessante aprender sobre Strings...:**



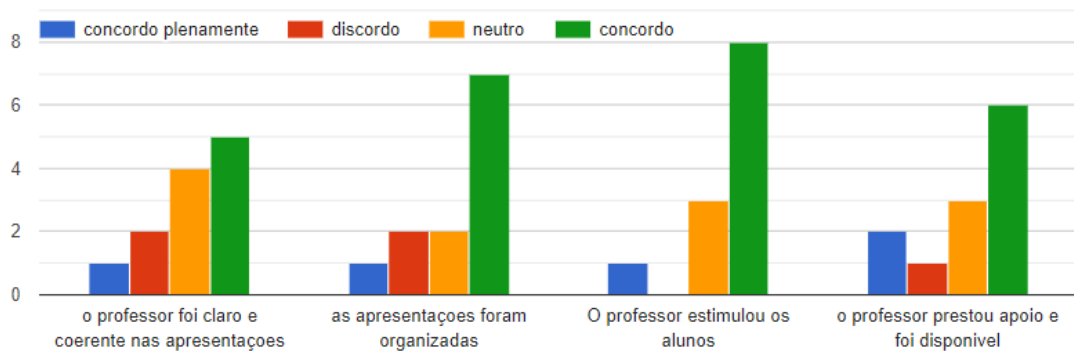


## 6. Nível das apresentações do professor e resposta aos alunos

a) No DEV C++



g) No Arduino



## 7. Respostas livres para a questão “Deixaria alguma sugestão para as aulas de Programação?”:

h) Resposta livre no final da 1ª parte:

- Precisa de explicar melhor a matéria
- Exemplos mais claros
- Mais divertidas e práticas
- Mais práticas e mais divertidas

i) Resposta livre no final da 2ª parte:

- Foi bom
- Exemplos mais claros e diretos

## Anexo G1 – Avaliação sumativa – Dimensão Atitudes e valores no DEV C++

| Aluno | Dia 3                         |                        |                        |                                                                                    | Dia 5                         |                        |                        |                                                                                 | MÉDIA FINAL |
|-------|-------------------------------|------------------------|------------------------|------------------------------------------------------------------------------------|-------------------------------|------------------------|------------------------|---------------------------------------------------------------------------------|-------------|
|       | Empenho nas tarefas propostas | Participação nas aulas | Iniciativa e autonomia | Observações                                                                        | Empenho nas tarefas propostas | Participação nas aulas | Iniciativa e autonomia | Observações                                                                     |             |
| I     | 5                             | 5                      | 5                      | Participativo e demonstra entender os conceitos da apresentação                    | 4                             | 4                      | 4                      | Participativo, mas não manifestou muitas dúvidas individuais...                 | 18          |
| H     | 4                             | 3                      | 5                      | Demonstra capacidades na resolução dos exercícios na distinção entre o scanf/fgets | 5                             | 4                      | 4                      | Demonstrou muito empenho e conseguiu um bom acompanhamento dos exercícios.      | 16,8        |
| C     |                               |                        |                        | faltou                                                                             | 4                             | 3                      | 4                      | A falta da aula anterior dificultou o acompanhamento, mas empenhou-se bastante. | 14,8        |
| D     | 4                             | 3                      | 4                      | Demonstra bons conhecimentos e capacidades.                                        | 5                             | 3                      | 5                      | Conseguiu realizar todos os exercícios, e manifestou interesse e boa autonomia  | 16,2        |

|   |   |   |   |                                                              |   |   |   |                                                                                                                         |      |
|---|---|---|---|--------------------------------------------------------------|---|---|---|-------------------------------------------------------------------------------------------------------------------------|------|
| A | 4 | 3 | 4 | Demonstra interesse.                                         | 5 | 3 | 4 | Demonstra empenho e ritmo estável.                                                                                      | 15,6 |
| G | 4 | 3 | 4 | O DEV C++ no seu PC não estava a permitir a depuração.       | 5 | 3 | 4 | Trabalha bastante com o aluno A e ambos colaboram com empenho nas tarefas e entreaajuda.                                | 15,6 |
| B | 5 | 5 | 5 | Participativa e demonstra interesse                          | 5 | 4 | 4 | Distraiu-se um pouco, e manifestou algumas dúvidas da aula anterior, mas pouca autonomia para o seguimento das tarefas. | 18,8 |
| M | 4 | 4 | 3 | Muito brincalhão e distrai-se um pouco                       | 3 | 4 | 4 | Tentou mostrar mais empenho, exercícios bastante atrasados.                                                             | 14,6 |
| K | 3 | 3 | 4 | Demonstra algum empenho                                      | 3 | 3 | 3 | Hoje ficou junto com o Miguel, os dois não se ouviram muito, mas mostraram algum interesse.                             | 12,6 |
| J | 4 | 3 | 4 | Manifestou algumas dificuldades básicas.                     | 3 | 3 | 3 | Tem um ritmo muito próprio, mas não participa muito                                                                     | 13,4 |
| E | 5 | 5 | 4 | Demonstra interesse, participativo e com vontade de aprender | 5 | 5 | 5 | Dirige-se frequentemente ao quadro para tirar duvidas das questões e mostrou ter assimilado a maioria com conceitos.    | 19,4 |

|   |   |   |   |                                                                                                    |   |   |   |                                                                                                                                        |      |
|---|---|---|---|----------------------------------------------------------------------------------------------------|---|---|---|----------------------------------------------------------------------------------------------------------------------------------------|------|
| F | 4 | 4 | 4 | Brincalhão mas dedicado e demonstra interesse                                                      | 5 | 4 | 4 | Mantem um ritmo de trabalho estável embora com varias dificuldades.                                                                    | 16,8 |
| L | 5 | 5 | 5 | Demonstra interesse, ficou o intervalo para perceber o exercício 1, com dificuldade nos ciclos for | 4 | 3 | 4 | Mantem um ritmo de trabalho estável embora com dificuldades significativas, talvez precise de exercícios de aprendizagem mais simples. | 17,4 |

## Anexo G2 – Avaliação sumativa – Dimensão Atitudes e valores no Arduino

| Grupo/ Alunos |   | Dia 10                        |                        |                                | Dia 12                        |                        |                                | NOTA FINAL |
|---------------|---|-------------------------------|------------------------|--------------------------------|-------------------------------|------------------------|--------------------------------|------------|
|               |   | Empenho nas tarefas propostas | Iniciativa e autonomia | Dinamismo no trabalho em grupo | Empenho nas tarefas propostas | Iniciativa e autonomia | Dinamismo no trabalho em grupo |            |
| A             | I | 5                             | 4                      | 5                              | 5                             | 4                      | 5                              | 18,8       |
|               | K | 3                             | 3                      | 4                              | 4                             | 4                      | 4                              | 14,6       |
| B             | H | 5                             | 4                      | 5                              | 5                             | 4                      | 5                              | 18,8       |
|               | C | 3                             | 3                      | 4                              | 4                             | 4                      | 4                              | 14,6       |
|               | F | 4                             | 3                      | 4                              | 4                             | 4                      | 4                              | 15,4       |
| C             | B | 5                             | 4                      | 5                              | 5                             | 4                      | 5                              | 18,8       |
|               | D | 4                             | 4                      | 5                              | 4                             | 4                      | 4                              | 16,6       |
| D             | M | 4                             | 4                      | 4                              | 5                             | 4                      | 5                              | 17,4       |
|               | A | 4                             | 4                      | 4                              | 5                             | 4                      | 5                              | 17,4       |
| E             | E | 5                             | 4                      | 5                              | 5                             | 4                      | 5                              | 18,8       |
|               | G | 5                             | 4                      | 5                              | 5                             | 4                      | 5                              | 18,8       |
| F             | L | 5                             | 4                      | 5                              | 5                             | 4                      | 5                              | 18,8       |
|               | J | 3                             | 3                      | 4                              | 4                             | 4                      | 4                              | 14,6       |

### Anexo G3 – Avaliação sumativa – Dimensão Conhecimento e capacidades cognitivas DEV C++

| Aluno | Questão |     |    |     |     |     |     |     |   |                 |
|-------|---------|-----|----|-----|-----|-----|-----|-----|---|-----------------|
|       | 1a      | 1b  | 2  | 3   | 4   | 5b  | 6a  | 6b  | 7 | Final<br>(0-20) |
| I     | 80      |     | 80 | 30  | 100 |     |     |     |   | 9,7             |
| H     | 80      | 80  |    |     | 100 |     |     |     |   | 8,8             |
| C     |         |     |    |     |     |     |     |     |   | 0               |
| D     | 100     |     | 80 | 100 |     | 100 | 100 | 100 |   | 11,4            |
| A     | 80      |     |    |     | 100 |     |     |     |   | 6,4             |
| G     | 80      |     | 80 | 30  | 100 |     |     |     |   | 9,7             |
| B     | 80      |     | 80 |     | 50  |     |     |     |   | 6,8             |
| M     | 80      |     | 80 | 100 | 50  |     |     |     |   | 9,8             |
| K     | 80      |     | 80 | 100 | 50  |     |     |     |   | 9,8             |
| J     | 80      |     | 90 | 100 | 50  |     |     |     |   | 10,1            |
| E     | 80      |     |    | 100 | 100 |     |     |     |   | 9,4             |
| F     | 80      | 80  |    | 100 | 100 |     |     |     |   | 11,8            |
| L     | 80      | 100 |    | 100 | 100 |     |     |     |   | 12,4            |

## **Anexo G4 – Avaliação sumativa – Dimensão Conhecimento e capacidades cognitivas Arduino**

### **Cotações**

| Componente                   | Cotação |
|------------------------------|---------|
| 1,2,3<br>(montagem e testes) | 20      |
| 4                            | 30      |
| 5                            | 20      |
| 6                            | 30      |
| Total                        | 100     |

| Grupo/Alunos |   | 1,2,3 | Observações | 4   | Observações                                                                                                                                    | 5  | Observações | 6   | Observações                                                       | Total (0-20) |
|--------------|---|-------|-------------|-----|------------------------------------------------------------------------------------------------------------------------------------------------|----|-------------|-----|-------------------------------------------------------------------|--------------|
| I            | I | 100   |             | 100 | Identificaram o problema de não estar a ser executada a montagem envio da string2 para o lcd, e o David identificou também uma solução viável. |    |             | 70  |                                                                   | 14,2         |
|              | K | 100   |             | 100 |                                                                                                                                                |    |             | 70  |                                                                   | 14,2         |
| II           | H | 100   |             | 100 |                                                                                                                                                |    |             | 70  | Demonstrou dificuldades, mas com um grande empenho para resolver. | 14,2         |
|              | C | 100   |             | 100 |                                                                                                                                                |    | 70          |     |                                                                   | 14,2         |
|              | F | 100   |             | 100 |                                                                                                                                                |    | 70          |     |                                                                   | 14,2         |
| III          | B | 100   |             | 100 |                                                                                                                                                |    |             | 100 |                                                                   | 16           |
|              | D | 100   |             | 100 |                                                                                                                                                |    |             | 100 |                                                                   | 16           |
| IV           | M | 100   |             | 100 |                                                                                                                                                | 50 |             | 50  |                                                                   | 15           |



|    |   |     |                                                                                                                                          |     |  |    |  |    |  |    |
|----|---|-----|------------------------------------------------------------------------------------------------------------------------------------------|-----|--|----|--|----|--|----|
|    | A | 100 |                                                                                                                                          | 100 |  | 50 |  | 50 |  | 15 |
| V  | E | 100 |                                                                                                                                          | 100 |  |    |  | 50 |  | 13 |
|    | G | 100 |                                                                                                                                          | 100 |  |    |  | 50 |  | 13 |
| VI | L | 100 | Pequenas dificuldades iniciais com a correspondência dos fios entre Arduino e os módulos mas com alguma orientação resolveram facilmente | 100 |  |    |  | 50 |  | 13 |
|    | J | 100 |                                                                                                                                          | 100 |  |    |  | 50 |  | 13 |

| Questão            | 1,2,3                                                                                                                         | 4                                                                                                                                                                                                                                             | 5 | 6                                                                                                                                                                                                      |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Observações gerais | a montagem decorreu sem problemas a apontar e rapidamente, apesar de com alguns grupos quase sem consultar o manual fornecido | A identificação da resolução da questão decorreu facilmente em todos os grupos. No entanto foi identificado um problema na execução, contraditório com o resultado esperado... que se resolveu facilmente e aplicou-se a correção para todos. |   | Os alunos entenderam a logica de receção e execução de comandos mas demonstraram dificuldades na implementação das expressões de condição compostas encadeadas - <code>if {} else{ if{} else{}}</code> |



## Anexo G5 – Avaliação sumativa – Dimensão Conhecimento e capacidades cognitivas Teste Final

| Aluno/<br>Questão | Questão |   |     |     |     |     |                 |
|-------------------|---------|---|-----|-----|-----|-----|-----------------|
|                   | 1       | 2 | 3   | 4   | 5   | 6   | Final<br>(0-20) |
| I                 | 100     | 0 | 100 | 100 | 100 | 100 | 16,6            |
| H                 | 0       | 0 | 100 | 100 | 0   | 100 | 10,2            |
| C                 | 100     | 0 | 0   | 0   | 0   | 100 | 6,4             |
| D                 | 100     | 0 | 100 | 0   | 100 | 100 | 13,2            |
| A                 | 100     | 0 | 100 | 0   | 100 | 0   | 9,8             |
| G                 | 100     | 0 | 100 | 100 | 0   | 100 | 13,2            |
| B                 | 100     | 0 | 0   | 0   | 0   | 100 | 6,4             |
| M                 | 100     | 0 | 100 | 0   | 100 | 0   | 9,8             |
| K                 | 100     | 0 | 0   | 0   | 0   | 100 | 6,4             |
| J                 | 0       | 0 | 100 | 0   | 0   | 100 | 6,8             |
| E                 | 100     | 0 | 100 | 100 | 0   | 100 | 13,2            |
| F                 | 0       | 0 | 0   | 0   | 0   | 100 | 3,4             |
| L                 | 100     | 0 | 100 | 100 | 0   | 100 | 13,2            |

| Questão | Cotação |
|---------|---------|
| 1       | 15      |
| 2       | 17      |
| 3       | 17      |
| 4       | 17      |
| 5       | 17      |
| 6       | 17      |
| Total   | 100     |

## Anexo G6 – Avaliação sumativa – Resultado Global

| Nome    | Atitudes e valores |          | Conhecimento e Capacidades cognitivas |         |             | Nota Final |
|---------|--------------------|----------|---------------------------------------|---------|-------------|------------|
|         | DEV C++            | Arduino  | DEV C++                               | Arduino | T. aferição |            |
| I       | 18                 | 18,8     | 9,7                                   | 14,2    | 16,6        | 15,64      |
| H       | 16,8               | 18,8     | 8,8                                   | 14,2    | 10,2        | 14,52      |
| C       | 14,8               | 18,8     | 0                                     | 14,2    | 6,4         | 11,88      |
| D       | 16,2               | 14,6     | 11,4                                  | 14,2    | 13,2        | 14,14      |
| A       | 15,6               | 15,4     | 6,4                                   | 14,2    | 9,8         | 12,85      |
| G       | 15,6               | 18,8     | 9,7                                   | 16      | 13,2        | 15,06      |
| B       | 18,8               | 16,6     | 6,8                                   | 16      | 6,4         | 14,05      |
| M       | 14,6               | 17,4     | 9,8                                   | 15      | 9,8         | 13,94      |
| K       | 12,6               | 17,4     | 9,8                                   | 15      | 6,4         | 13,1       |
| J       | 13,4               | 18,8     | 10,1                                  | 13      | 6,8         | 13,35      |
| E       | 19,4               | 18,8     | 9,4                                   | 13      | 13,2        | 15,35      |
| F       | 16,8               | 18,8     | 11,8                                  | 13      | 3,4         | 14,2       |
| L       | 17,4               | 17,18462 | 12,4                                  | 13      | 13,2        | 15,04      |
| Média 1 | 16,2               | 17,7     | 8,9                                   | 14,2    | 9,9         | 14,1       |
| Média 2 | 16,9               |          | 11,0                                  |         |             |            |

| Componente                                           | Pêso |
|------------------------------------------------------|------|
| a) Dev c++ - Atitudes e valores + Aptidões           | 25   |
| b) Arduino - Atitudes e valores + Aptidões           | 25   |
| c) Conhecimento e capacidades cognitivas Dev c++     | 20   |
| d) Conhecimento e capacidades cognitivas Arduino     | 20   |
| e) Conhecimento e capacidades cognitivas teste final | 10   |
| Total                                                | 100  |

## **Anexo H – Manual de conceitos teóricos**

**Universidade de Lisboa**

**Escola secundária de Sacavém**



AGRUPAMENTO DE ESCOLAS  
EDUARDO GAGEIRO

**Curso profissional Técnico de Gestão e Programação de  
Sistemas Informáticos**

**Ensino de variáveis estruturadas String em linguagem C  
com recurso à plataforma Arduino**

**Manual das aulas - parte teórica**

**Intervenção pedagógica supervisionada**

**2019/20**

**Formando:**

**Paulo de Oliveira Ferreira**

## Conteúdo

|     |                                            |   |
|-----|--------------------------------------------|---|
| 1   | Introdução.....                            | 3 |
| 2   | Definição de String? .....                 | 3 |
| 2.1 | Porquê o “\0”? .....                       | 4 |
| 3   | Outra forma de inicializar uma String..... | 4 |
| 4   | Diferença entre “A” e ‘A’ .....            | 5 |
| 5   | Strings e ponteiros para Strings.....      | 5 |
| 6   | Funções sobre Strings.....                 | 6 |

## 1 Introdução

Até agora foram utilizadas variáveis de valor único, de estrutura simples, como Int, Float, Char, etc.. Na linguagem C existem também as variáveis estruturadas, sendo uma delas as Strings.

Inevitavelmente, para se falar de Strings, é necessário abordar o conceito de array, que por sua vez será abordado mais detalhadamente na temática seguinte da disciplina.

Array é um vetor (conjunto ordenado) de elementos do mesmo tipo.

### Exemplo 1:

Presumindo que nosso Drone tem uma função para obter e guardar a altitude a cada 5 segundos. Não seria nada prático usar uma variável "int altitude\_n;" para cada leitura... Então numa só variável, do tipo array de inteiros, podemos guardar uma série delas identificáveis pela sua posição nesse vetor.

```
//Array de inteiros com capacidade para 1000 inteiros.
```

```
int drone_altitudes_recolhidas[1000];
```

### Exemplo 2:

```
char nome[5];
```

//Este array de caracteres, cujo conteúdo ainda não é definido tem capacidade para 5 caracteres.

Mas, continuando com este exemplo de array de caracteres, o que acontece quando de seguida lhe atribuímos um valor da forma que se segue?

```
Nome = "Ana";
```

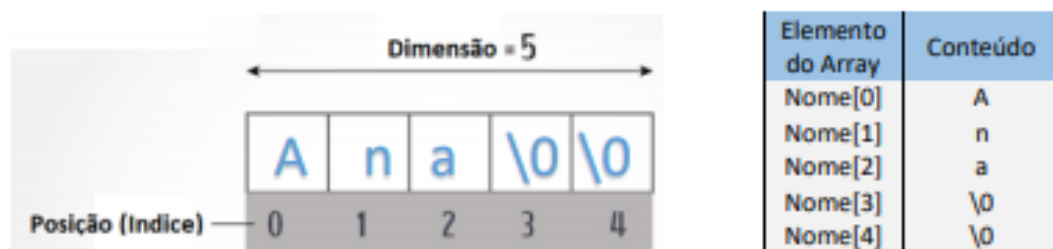


Figura 1- Estrutura de um array cujo conteúdo é uma String de caracteres

A partir do momento em que atribuímos uma string de caracteres como valor a esta variável, deixamos de ter apenas um array de caracteres para termos uma variável String, porque a seguir ao último carácter foi explicitamente colocado o elemento '\0' indicando o final da String de caracteres. Em todos os outros elementos que se seguem é também colocado um '\0', que alias já existia antes de o conteúdo deste array ser inicializado. Veremos a seguir o porquê e significado deste elemento terminador '\0' neste contexto.



Podemos aceder a cada elemento deste vetor de caracteres, que agora é uma String, individualmente a cada caracter, ou pelo seu conjunto como variável String:

```
printf("O nome do aluno é: %s", Nome);
```

// Como podes ver, a letra do operador que já conhecias "%s", refere-se a String

```
printf("A primeira letra do nome é : %c", Nome[0]);
```

//Neste caso a letra do operador que também já conhecias "%c", refere-se a um caracter, mais propriamente na posição/índice 0 do array.

## 2 Definição de String?

Uma String é definida como uma variável estruturada na forma de Array do tipo char. Ou seja um conjunto de caracteres armazenados num vetor cuja única diferença para um simples array é o facto de a seguir ao último caracter existir o "\0" (elemento zero da tabela ASCII), e que é atribuído automaticamente quando a String é inicializada ou alterada.

Ou seja é um Array de n caracteres + 1. Por isso a dimensão de uma String convém ser sempre superior ou igual ao número de caracteres necessários +1. Caso contrário os restantes caracteres serão ignorados e não guardados.

### 2.1 Porquê o "\0"?

O caracter terminador de uma String poderia ser qualquer elemento da tabela ASCII, no entanto é necessário evitar problemas que resultariam do facto de o caracter terminador ser um caracter que faça parte do nosso alfabeto. Ou seja decidiu-se utilizar o elemento zero da tabela ASCII, designado NULL, que representa o "vazio", "não atribuído", ou neste caso "fim de String". É utilizado em várias situações da programação em C assim como noutras linguagens. Não confundir o "\0" com o caracter "0", na verdade não é um caracter, mas apenas um dos elementos da tabela ASCII com uma função de controlo e representado no código da linguagem C pelo símbolo "\0".

## 3 Outra forma de inicializar uma String:

Atrás vimos que é possível declarar uma vetor de caracteres definindo apenas a sua dimensão (char nome[5];). Ou seja, neste caso não foi ainda inicializado o seu conteúdo e por isso não podemos ainda chamar-lhe variável String.

Mas podemos criar e declarar uma String sem indicar diretamente a sua dimensão, inicializando o seu conteúdo:

```
char Str4[] = "Arduino";
```

Ou seja, quando a dimensão não é explicitamente definida, é obrigatório ser imediatamente inicializada (atribuir-lhe um valor). E neste caso a dimensão assumida será a desse "valor" atribuído mais o elemento terminador '\0':

A variável Str4 irá então conter:

7 Carateres ("Arduino") + 1 elemento terminador (\0) = 8 elementos.

Isto implica que após a inicialização, se tentarmos alterar o conteúdo de Str para um valor com dimensão superior a 7, como por exemplo:

*Strcpy(Str, "Ardumotic"); // copiar a palavra "Ardumotic" para a variável Str*

Neste caso apenas seria colocado na variável Str o conteúdo "Ardumot". Todos os restantes caracteres são ignorados. Por isso pensa sempre na melhor forma de inicializar uma String e na respetiva dimensão necessária.

#### 4 Diferença entre "A" e 'A'

Em C, as Strings são representadas utilizando aspas, enquanto os caracteres são representados utilizando plicas. Neste caso apesar de o número de caracteres ser o mesmo, enquanto o carater 'A' precisa apenas de 8 bits de memória, a String "A" precisa de 16bits de memória, por contém também o elemento terminador \0;

*Tabela 1 - Representação do conteúdo em memória de uma variável String e uma variável caracter.*

| char letra1='A'<br>char string1[]="A"; | Endereço | Conteúdo | Variável |
|----------------------------------------|----------|----------|----------|
|                                        | ...      | ...      | ...      |
|                                        | 2999     | A        | letra1   |
|                                        | 3000     |          |          |
|                                        | 3001     | A        | string1  |
|                                        | 3002     | \0       |          |
|                                        | 3003     |          |          |
|                                        | 3004     |          |          |
|                                        | 3005     |          |          |

## 5 Strings e ponteiros para Strings

O conceito de ponteiro é também um conceito incontornável nesta temática, pois ele está implícito por exemplo em algumas funções que retornam Strings pelo seu endereço e não pelo conteúdo. O que tem a vantagem de tornar a execução dos programas mais rápida e eficiente.

Vejamos o exemplo com a seguinte sequência:

*Tabela 2 - Representação e distinção entre Strings e ponteiros para Strings*

| char Str1[]="ABC"; |          |          | char Str1[]="ABC";<br>char* str2; |          |          | char Str1[]="ABC";<br>char* str2;<br>str2=str1; |          |          |
|--------------------|----------|----------|-----------------------------------|----------|----------|-------------------------------------------------|----------|----------|
| Endereço           | Conteúdo | Variável | Endereço                          | Conteúdo | Variável | Endereço                                        | Conteúdo | Variável |
| ...                | ...      | ...      | ...                               | ...      | ...      | ...                                             | ...      | ...      |
| 2999               | A        | Str1     | 2999                              | A        | Str1     | 2999                                            | A        | Str1     |
| 3000               | B        |          | 3000                              | B        |          | 3000                                            | B        |          |
| 3001               | C        |          | 3001                              | C        |          | 3001                                            | C        |          |
| 3002               |          |          | 3002                              |          |          | 3002                                            |          |          |
| 3003               |          |          | 3003                              | NULL     | str2     | 3003                                            | 2999     | str2     |
| 3004               |          |          | 3004                              |          |          | 3004                                            |          |          |
| 3005               |          |          | 3005                              |          |          | 3005                                            |          |          |

As seguintes duas instruções têm exatamente o mesmo output, ou seja, em termos de acesso ao conteúdo, quer a String quer o ponteiro para String podem ser interpretados da mesma forma quando usamos o printf.

```
printf("%s",str1); // output: ABC
```

```
printf("%s",str2); //output: ABC
```

Observar que, no caso do ponteiro \*str2, para acedermos ao conteúdo da localização apontada chamamos a variável str2 sem o asterisco;

Umas das vantagens dos ponteiros é que podemos alterar o conteúdo simplesmente alterando o endereço para onde aponta. Ou no caso das funções, quando se espera uma String no retorno, passar apenas o endereço dessa String, tornando a execução mais rápida:

*Tabela 3 - Alteração do conteúdo de uma String pelo seu endereço de memória.*

```
char str3[]="DEF";
```

```
strcpy(str2,str3);
```

| Endereço | Conteúdo | Variável |
|----------|----------|----------|
| ...      | ...      | ...      |
| 2999     | A        | str1     |
| 3000     | B        |          |
| 3001     | C        |          |
| 3002     |          |          |
| 3003     | 3006     | str2     |
| 3004     |          |          |
| 3005     |          |          |
| 3006     | D        | str3     |
| 3007     | E        |          |
| 3008     | F        |          |

## 6 Funções sobre Strings

As funções da biblioteca `string.h` permitem executar operações comuns sobre Strings de caracteres terminadas em NULL (“\0”). Em algumas delas não é necessário declarar a inclusão da biblioteca `string.h`.

Apresentam-se de seguida a mais utilizadas e que será abordadas nestas aulas.

*Tabela 4 - Funções principais de manipulação de Strings na linguagem C.*

| Retorno | Estrutura (nome e parâmetros)                        | Descrição                                   |
|---------|------------------------------------------------------|---------------------------------------------|
| char *  | <code>strcat (char *, const char *)</code>           | Concatenação de duas Strings                |
| char *  | <code>strchr (const char *, int)</code>              | Localizar um caracter numa String           |
| int     | <code>strcmp (const char *, const char *)</code>     | Comparar duas Strings                       |
| char *  | <code>strcpy (char *, const char *)</code>           | Copia de Strings                            |
| int     | <code>strcasecmp (const char *, const char *)</code> | Compara duas Strings ignorando Case         |
| char *  | <code>strcasestr (const char *, const char *)</code> | Procura a primeira ocorrência de S2 em S1   |
| size_t  | <code>strlen (const char *)</code>                   | Retorna o comprimento da String             |
| char *  | <code>strlwr (char *)</code>                         | Converte a string para LowerCase            |
| char *  | <code>strrev (char *)</code>                         | Retorna a string invertida                  |
| char *  | <code>strstr (const char *, const char *)</code>     | Localiza um substring                       |
| char *  | <code>strtok (char *, const char *)</code>           | Faz o split da String pelo carater indicado |
| char *  | <code>strupr (char *)</code>                         | Converte a String um UperCase               |



# ANEXO I – Manual e Guião da atividade Ardumotic



Escola Secundária de Sacavém  
Curso profissional Técnico de Gestão e  
Programação de Sistemas Informáticos  
Professor: Paulo Ferreira  
(Prática de Ensino Supervisionada)

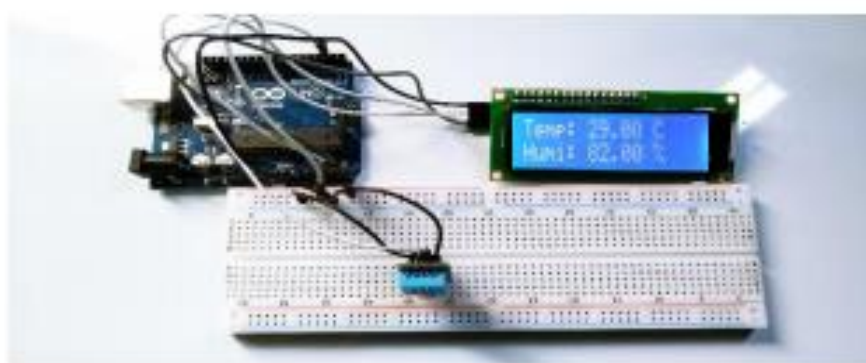
**Disciplina:** Programação e Sistemas de Informação - 10º Ano  
**Módulo 4 –** Estruturas de Dados Estáticas - Strings

**Data:** 03/03/2020

**Aluno**

**Nome:** \_\_\_\_\_ **Nº:** \_\_\_\_\_

## *Atividade Ardumotic*



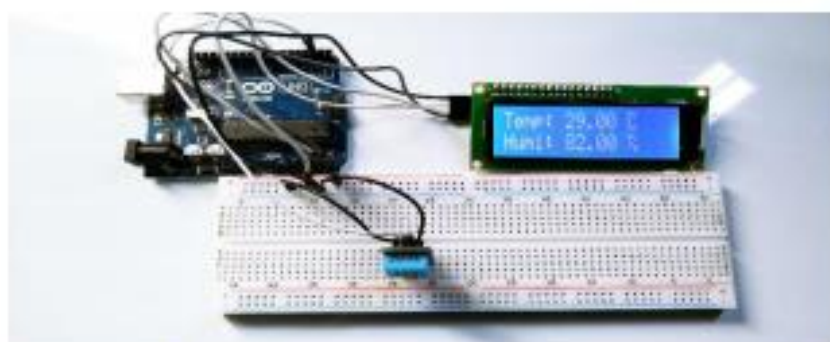
## Conteúdo

|     |                                                                                     |    |
|-----|-------------------------------------------------------------------------------------|----|
| 1   | Apresentação .....                                                                  | 2  |
| 2   | Montagem e teste do Módulo DHT11.....                                               | 3  |
| 2.1 | Ligação entre o Arduino e o módulo DHT11 (sensores de temperatura e humidade). .... | 3  |
| 2.2 | Adicionar bibliotecas ao Arduino.....                                               | 3  |
| 2.3 | Código de teste e leitura dos sensores de temperatura e humidade .....              | 3  |
| 3   | Montagem e teste do Display LCD .....                                               | 4  |
| 3.1 | Ligações .....                                                                      | 4  |
| 3.2 | Teste do LCD e script .....                                                         | 5  |
| 4   | Visualização da temperatura e humidade no LCD .....                                 | 6  |
| 4.1 | Exercício .....                                                                     | 9  |
| 5   | Comunicação entre Arduino e Monitor-Série.....                                      | 9  |
| 5.1 | Exercício: .....                                                                    | 10 |
| 6   | Envio de comandos via Monitor Serie .....                                           | 11 |
| 6.1 | Exercício: .....                                                                    | 13 |

## 1 Apresentação

Esta atividade insere-se no programa da disciplina Programação e Sistemas de Informação, no Módulo 4 – Estruturas de Dados Estáticas. Pretende-se com esta atividade aplicar os conhecimentos adquiridos da temática Strings num cenário de aplicação concreta e observável, utilizando para isso a plataforma Arduino.

Este guião orienta a elaboração de um sistema de captação de sensores de temperatura e humidade e visualização dessa informação num display LCD.



*Figura 1 - Montagem e ligação completa*

A seguir à montagem, as partes constituintes da atividade são:

- A captação dos sinais dos sensores através do Módulo DHT11, que incorpora os dois sensores, e permite uma fácil leitura com as suas funções de API predefinidas na sua biblioteca.
- A visualização de informação no display através do módulo LiquidCrystal\_I2C, que também disponibiliza uma biblioteca com funções predefinidas e de utilização direta para receção e projeção de Strings no LCD.
- Para complementar a funcionalidade desta montagem, o Arduino contém na sua IDE a funcionalidade Monitor-Serie (MS) que assenta também na transferência de Strings bidireccionalmente, entre o Arduino e o computador ao qual esta ligado via Serial-COM. Esta funcionalidade é útil para:
  - Testes e observação do funcionamento dos módulos (Debug), através do envio de Strings (para o MS) com valores de variáveis internas do programa.
  - Envio de comandos, do MS para o Arduino, através de Strings estruturadas, para por exemplo acender ou apagar a luz de fundo do LCD.

## 2 Montagem e teste do Módulo DHT11

### 2.1 Ligação entre o Arduino e o módulo DHT11 (sensores de temperatura e humidade).

Faça as ligações entre o Arduino e o módulo DHT11 como ilustrado na figura seguinte e respeitando a correspondência entre os pinos de ligação da tabela:

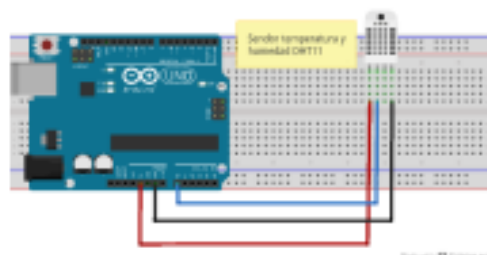


Figura 2 - Ligação entre o Arduino e o módulo DHT11

Tabela 1 - Pinos de ligação e correspondência entre o Arduino e o módulo DHT11

| DHT11 | Arduino uno |
|-------|-------------|
| "+"   | 5v          |
| "-"   | GND         |
| "s"   | A0          |

### 2.2 Adicionar bibliotecas ao Arduino

No menu Arduino seleccionar:

- Rascunho/Script-> importar biblioteca-> adicionar biblioteca
- Seleccionar pasta DHT
- Repetir este processo para a pasta "LiquidCrystal\_I2C"

Quando aceder novamente a este menu deverá observar na lista a existência destas bibliotecas.

### 2.3 Código de teste e leitura dos sensores de temperatura e humidade

A leitura dos sensores e a conversão de valores nas respetivas unidades de temperatura e humidade é feita automaticamente pela função fornecida na livreria do módulo DHT11 (`#include "dht.h"`):

```
DHT.read11(dht_apin);
```

Onde *dht\_apin* é o pino de entrada analógica do Arduino onde estará ligado o sinal elétrico de output "s" do módulo DHT11, neste caso é usada a entrada analógica A0.





Na execução do código exemplo a seguir é possível fazer essa leitura e enviar os respetivos valores via Serial-COM para o Monitor-Serie do Arduino;

O seguinte programa demonstra a utilização das funções de leitura da temperatura e humidade do módulo DHT11. Esses valores são enviados via Serial-Com para o Monitor Serie do Arduino.

```
//-----
#include "dht.h"
#define dht_apin A0 // pino de entrada analógica utilizado

dht DHT; // Cria um objeto do tipo DHT que permite usar as funções que
controlam o módulo DHT11 e captação dos sensores

void setup() { //Configurações iniciais executadas antes do Loop principal
 Serial.begin(9600);
 delay(500); // milissegundos de tempo de espera para deixar o sistema
 arrancar
 Serial.println("Teste DHT11 – Modulo de temperatura e humidade\n\n");
 delay(1000); //esperar um pouco antes de aceder ao modulo DHT11
} //Final do "setup()"
//-----
void loop() { //função principal

 DHT.read11(dht_apin); //ler sensores de temperatura e humidade
 //enviar informação via Monitor Serie
 Serial.print("Humidade atual = ");
 Serial.print(DHT.humidity);
 Serial.print("% ");
 Serial.print("temperatura = ");
 Serial.print(DHT.temperature);
 Serial.println("C ");

 delay(2000); //esperar 2 segundos para a próxima medição.
}
```

### 3 Montagem e teste do Display LCD

#### 3.1 Ligações

Para a montagem do display LCD IIC I2C TWI Serial 1602 16x2 com o Arduino, faça a ligação dos 4 fios como indicado na tabela, dois são para alimentação e os outros dois são para comunicação:



Figura 3 - montagem e ligação entre o Arduino e o módulo IIC I2C

| I2C LCD1602 | SunFounder Uno |
|-------------|----------------|
| GND         | GND            |
| VCC         | 5V             |
| SDA         | A4             |
| SCL         | A5             |

Figura 4 - Ligações e respetivos pinos entre o Arduino e o módulo LCD IIC I2C

#### 3.2 Teste do LCD e script

Para o funcionamento do display LCD execute os passos seguintes:

- Criar um sketch no Arduino com o código fornecido abaixo.
- Fazer Upload do programa seguinte para o Arduino;
- Deverá ver-se o display I2C LCD1602 mostrar os seguintes caracteres: "GPSI" na linha 1 e "Arduotic" na linha 2.
- Analisar a sequência de execução de código observando o output.

O script apresentado a seguir permite enviar duas Strings para o LCD, uma para cada linha. Caso as Strings sejam superiores a 16 caracteres apenas aparecem os primeiros 16.

Teste o seu output para verificar o seu funcionamento.

```
//-----
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
/*****
char str1[]="GPSI"; //String a imprimir no LCD na linha 1
char str2[]="Arduotic!"; // String a imprimir no LCD na linha 2
*****/
```



```
// initialize the library with the numbers of the interface pins
LiquidCrystal_I2C lcd(0x27,16,2); // set the LCD address to 0x27 for a 16 chars
and 2 line display
/*****/
void setup()
{
 lcd.init(); //inicializar o lcd
 lcd.backlight(); //acender a luz d
}
/*****/
void loop()
{
 //Enviar string para a linha 1
 lcd.setCursor(0,0); // set the cursor to column 15, line 0
 lcd.print(str1); // Print a message to the LCD.
 //Enviar string para a linha 2
 lcd.setCursor(0,1); // set the cursor to column 15, line 1
 lcd.print(str2); // Print a message to the LCD.

 delay(1000); //wait for 250 microseconds
}
```

#### 4 Visualização da temperatura e humidade no LCD

O programa seguinte permite fazer uma concatenação de uma String base "temperatura: " com o valor da temperatura recebido do módulo DHT11 convertido em String.

Este programa inclui uma nova função "lcd\_msg\_scroll\_builder()" que permite usar Strings maiores que 16 caracteres, fazendo um scroll left.

A String resultante é enviada para a linha correspondente do LCD.

Executa e testa o seguinte programa:

O código completo do script para visualização de teste do funcionamento dos módulos DHT11 e LCD I2C é o seguinte. Podes copiar e executar o seguinte programa:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <dht.h>
#define printByte(args) write(args);

//variáveis e definições principais para as duas Strings e correspondentes linhas do lcd
char string1[200]="Temperatura: ";
char string1_completa[200]; //apos as concatenações, exemplo: "Temperatura:
23C"
```



---

```
char LCD_string_linha1[16]; // String final a imprimir no LCD, linha 1, os 16
carateres que irão aparecer
int lcd_str1_start_position=0; //início da substring quando string1_completa maior
que 16 carateres

char string2[200]="Humidade: ";
char string2_completa[200]; //apos as concatenações, exemplo: "Temperatura:
23C"
char LCD_string_linha2[16]; // String final a imprimir no LCD, linha 2, os 16
carateres que irão aparecer
int lcd_str2_start_position=0; //início da substring quando string1_completa maior
que 16 carateres

int frequencia_lcd=200; // tempo de pausa em cada loop (milissegundos),
controlando a velocidade de atualização e do scroll left do lcd do LCD
LiquidCrystal_I2C lcd(0x27,16,2); // cria um objeto LCD e define os seus
parâmetros: address 0x27, 16 colunas e 2 duas linhas

#define dht_apin A0 // modulo dht11-pino de entrada analógica utilizado pe
dht DHT; // cria um objeto do tipo DHT que permite usar as respetivas funções
que controlam o modulo DHT11
/*****
*****/
void setup() { //Configurações iniciais executadas antes do Loop principal
 lcd.init(); //initialize the lcd
 lcd.backlight(); //open the backlight
}
/*****
*****/
void loop(){

 DHT.read11(dht_apin); //ler sensores de temperatura e humidade
 //conversão de valores de temperatura e humidade para Strings

 int temperatura=DHT.temperature;
 int humidade=DHT.humidity;
 char str_temperatura[3];
 char str_humidade[3];
 sprintf(str_temperatura,"%i", temperatura); //converte o valor inteiro de
temperatura par string
 sprintf(str_humidade,"%i", humidade); //converte o valor inteiro de humidade
par string

 //-----envio de informação para o LCD-----
 lcd.clear(); //limpar o lcd
 //construir a string completa da linha 1
```



```
strcpy(string1_completa,string1);//reinicia o conteudo da string1_completa
com o conteudo da string1
strcat(string1_completa,str_temperatura);//adicionar o valor da temperatura
em formato string à string_Completa
strcat(string1_completa,"C"); //adicionar o símbolo de temperatura em graus
celcius
//construir a string completa da linha2
//??
//??
//??
//enviar as duas linhas para o lcd
lcd_msg_scol_builder();//caso as strings contenham mais de 16 carateres,
constroi as subtrings a mostrar atrves de um scol-left
lcd.setCursor(0,0); // colocar o cursor na linha 0, coluna 0
lcd.print(LCD_string_linha1); // enviar a linha1 para o LCD.
lcd.setCursor(0,1);//colocar o cursor na linha 1 coluna 0
lcd.print(LCD_string_linha2); // enviar a linha 2 para o LCD.

delay(frequencia_lcd);//esperar "Frenquencia_lcd" milisegundos para a
próxima ciclo.
}
/*****
*****/
//função para moldar a mensagem que aparece no lcd quando a dimensao da string
enviada é maior que a capacidade do lcd, fazendo um scroll left
void lcd_msg_scol_builder(){
//substring para a linha 1 - caso a string completa contenha mais que 16 carteres
if(strlen(string1_completa)>16){
strcpy(LCD_string_linha1, string1_completa + lcd_str1_start_position, 16);
if(lcd_str1_start_position<(strlen(string1_completa))){
lcd_str1_start_position++;
}
}
else{
lcd_str1_start_position=0;
}
}
else{
strcpy(LCD_string_linha1,string1_completa);
}
//substring para a linha 2 - caso a string completa contenha mais que 16
carteres
if(strlen(string2_completa)>16){
strcpy(LCD_string_linha2, string2_completa + lcd_str2_start_position, 16);
if(lcd_str2_start_position<strlen(string2)){
lcd_str2_start_position++;
}
}
else{
```

A completar



```

 lcd_str2_start_position=0;
 }
 else{
 strcpy(LCD_string_linha2,string2_completa);
 }
}

```

#### 4.1 Exercício

Complete a parte assinalada que falta no programa “?????????????” para construir a String completa da linha 1 do LCD, tal como para a linha 0.

### 5 Comunicação entre Arduino e Monitor-Série

Na imagem seguinte podemos observar uma forma de comunicação entre o Arduino e o Monitor-Série.

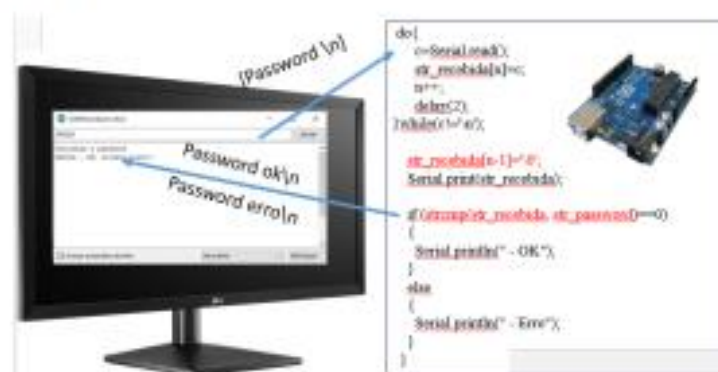


Figura 5- Comunicação entre Monitor-Série e Arduino

Através deste sketch podemos testar a comunicação entre o Monitor-Série e o Arduino da seguinte forma:

- É enviado um conjunto de caracteres (uma palavra) a partir do monitor série, para o Arduino;
- O Arduino recebe esse conjunto de caracteres, um a um, via Serial-Com, “montando” assim uma String com a mensagem completa;
- A seguir é feita uma comparação entre a String recebida com outra String (variável global) já definida no programa;
- Se ambas corresponderem, ou não, e usando para isso a função <strcmp>, que permite compara-las, então irá enviar uma resposta para o Monitor Série confirmando esse resultado de comparação.

O código completo deste script é o seguinte. Podes copiar e executar analisando o resultado:

```
char str_password[] = "GPSI20";
char str_recebida[100];
char c;
int n=0;
void setup()
{
 Serial.begin(9600);
 Serial.println("Introduza a password");
}
void loop()
{
 n=0;

 if(Serial.available())
 {
 do{
 c=Serial.read();
 str_recebida[n]=c;
 n++;
 delay(2); //Delay para o Arduino ter tempo de atualizar o buffer de receção
 }while(c!='\n');
 str_recebida[n-1]='\0';
 Serial.print("password recebida");
 if (strcmp(str_recebida,str_password)==0)
 {
 Serial.println(" - OK, tiveste sorte!!");
 }
 else
 {
 Serial.println(" - Password errada");
 }
 }
}
```

### 5.1 Exercício:

- Adiciona uma variável global (no início do programa) do tipo boolean chamada *login* e inicializada a *false*.
- Na parte do código assinalada, onde é feita a comparação das duas Strings, e consoante sejam iguais, ou não, faça a atribuição do valor *true* ou *false* à variável *login*.

## 6 Envio de comandos via Monitor Serie

O programa seguinte conjuga o programa do ponto 3.2 ("Teste do Display LCD") com o programa do ponto 5 ("Comunicação Arduino < - > Monitor Série").

Exemplo:

Se enviarmos a String "setstring2 PSI" via monitor serie, a função *strtok* vai isolar a primeira palavra identificando o comando, e de seguida isolando a palavra seguinte, vai-se redefinir a String da linha 1 do LCD para "PSI". Nas imagens seguintes é ilustrado o envio comado e respetivo resultado.

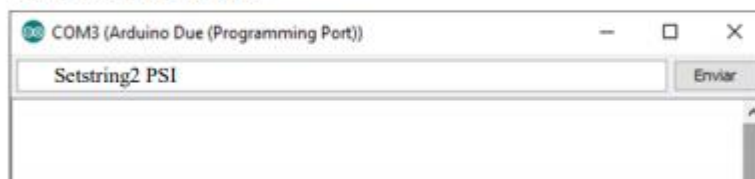


Figura 6 - Monitor-Serie com envio e comando

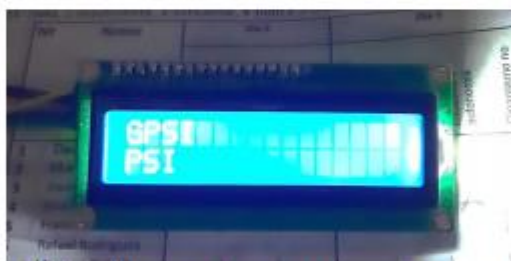


Figura 7 – Resultado do envio de comando para alterar mensagem do LCD na segunda linha.

Implemente e teste o programa, e teste também os dois comandos já definidos.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
/*****
char str1[100]="GPSI"; //String a imprimir no LCD na linha 1
char str2[]="Ardumotic!sempre"; // String a imprimir no LCD na linha 2
char str_recebida[100];
char c;
int n=0;
bool lcd_backlight=true;

// initialize the library with the numbers of the interface pins
```



```
LiquidCrystal_I2C lcd(0x27,16,2); // set the LCD address to 0x27 for a 16 chars
and 2 line display
/*****/
void setup()
{
 Serial.begin(9600);
 lcd.init(); //inicializar o lcd
 lcd.backlight(); //acender a luz d
}
/*****/
void loop()
{
 n=0;
 if(Serial.available()) //se existir algo a receber na serial com
 {
 do{//recebe um carater de cada vez
 c=Serial.read();
 str_recebida[n]=c;
 n++;
 delay(2); //Delay para o Arduino ter tempo de atualizar o buffer de receção
 }while(c!='\n');
 str_recebida[n]='\0';
 Serial.print("string recebida: ");
 Serial.println(str_recebida);
 //obter o comando através do split da string recebida, com a função strtok
 char* token = strtok(str_recebida, " "); //obtem o primeiro token

 if(strcmp(token,"setstring1")==0){
 token = strtok(NULL, " "); //obtem o segundo token
 strcpy(str1,token); //atribui o valor de token a String str1 correspondente à
 linha 0 do lcd
 }
 else {
 if(strcmp(token,"backlight")==0){
 if (lcd_backlight==true){
 lcd.noBacklight(); //apaga a luz do LCD
 lcd_backlight=false;
 }
 else{
 lcd.backlight(); //acende a luz do LCD
 lcd_backlight=true;
 }
 }
 }
 }
}

//-----//
//enviar strings para o lcd, linha 0 e linha 1 respetivamente
```

Split da String  
recebida e  
identificação do  
comando.



```
lcd.clear();

lcd.setCursor(0,0); //posiciona o cursos na linha 0 coluna 0
lcd.print(str1); // Imprime um String no LCD, na posição definina.

lcd.setCursor(0,1); // posiciona o cursos na linha 1 coluna 0
lcd.print(str2); // Envia uma string para visualizar no LCD, na posição definida.
delay(1000); //
}
```

### 6.1 Exercício:

Acrescente um comando para alterar a String visualizada na primeira linha do LCD, tal como mostrado nas imagens anteriores.

Exemplo: "setstring1 Programacao".

## ANEXO J – Tabela ASCII



Escola Secundária de Sacavém  
Curso profissional Técnico de Gestão e  
Programação de Sistemas Informáticos  
Professor: Paulo Ferreira  
(Prática de Ensino Supervisionada)

### Tabela ASCII

#### Elementos de controlo – não imprimíveis

| Bin       | Oct | Dec | Hex | Abrev | Notação com circunflexo | Código escape | Nome                                                                   |
|-----------|-----|-----|-----|-------|-------------------------|---------------|------------------------------------------------------------------------|
| 0000 0000 | 000 | 00  | 00  | NUL   | ^@                      | \0            | Nulo (inglês Nul)                                                      |
| 0000 0001 | 001 | 01  | 01  | SOH   | ^A                      |               | Início de cabeçalho (inglês Start of Header)                           |
| 0000 0010 | 002 | 02  | 02  | STX   | ^B                      |               | Início de texto (inglês Start of Text)                                 |
| 0000 0011 | 003 | 03  | 03  | ETX   | ^C                      |               | Fim de texto (inglês End of Text)                                      |
| 0000 0100 | 004 | 04  | 04  | EOT   | ^D                      |               | Fim de transmissão (inglês End of Transmission)                        |
| 0000 0101 | 005 | 05  | 05  | ENQ   | ^E                      |               | Consulta; inquirição (inglês Enquiry)                                  |
| 0000 0110 | 006 | 06  | 06  | ACK   | ^F                      |               | Confirmação (inglês Acknowledge)                                       |
| 0000 0111 | 007 | 07  | 07  | BEL   | ^G                      | \a            | Campanha; sinal sonoro (inglês Bell)                                   |
| 0000 1000 | 010 | 08  | 08  | BS    | ^H                      | \b            | Espaço atrás; retorno de 1 caractere (inglês Back-space)               |
| 0000 1001 | 011 | 09  | 09  | HT    | ^I                      | \t            | Tabulação horizontal (inglês Horizontal Tabulation)                    |
| 0000 1010 | 012 | 10  | 0A  | LF    | ^J                      | \n            | Alimentação de linha; mudança de linha; nova linha (inglês Line Feed)  |
| 0000 1011 | 013 | 11  | 0B  | VT    | ^K                      | \v            | Tabulação vertical (inglês Vertical Tabulation)                        |
| 0000 1100 | 014 | 12  | 0C  | FF    | ^L                      | \f            | Alimentação de formulário (inglês Form Feed)                           |
| 0000 1101 | 015 | 13  | 0D  | CR    | ^M                      | \r            | Retorno do carro; retorno ao início da linha (inglês Carriage Return)  |
| 0000 1110 | 016 | 14  | 0E  | SO    | ^N                      |               | Mover para fora; deslocamento para fora (inglês Shift Out)             |
| 0000 1111 | 017 | 15  | 0F  | SI    | ^O                      |               | Mover para dentro; deslocamento para dentro (inglês Shift In)          |
| 0001 0000 | 020 | 16  | 10  | DLE   | ^P                      |               | escape do enlace de dados; escape de conexão (inglês Data-Link Escape) |
| 0001 0001 | 021 | 17  | 11  | DC1   | ^Q                      |               | Controlo de dispositivo 1 (inglês Device Control 1)                    |
| 0001 0010 | 022 | 18  | 12  | DC2   | ^R                      |               | Controlo de dispositivo 2 (inglês Device Control 2)                    |
| 0001 0011 | 023 | 19  | 13  | DC3   | ^S                      |               | Controlo de dispositivo 3 (inglês Device Control 3)                    |
| 0001 0100 | 024 | 20  | 14  | DC4   | ^T                      |               | Controlo de dispositivo 4 (inglês Device Control 4)                    |
| 0001 0101 | 025 | 21  | 15  | NAK   | ^U                      |               | Confirmação negativa (inglês Negative-Acknowledge)                     |
| 0001 0110 | 026 | 22  | 16  | SYN   | ^V                      |               | Estado ocioso síncrono; espera síncrona (inglês Synchronous idle)      |
| 0001 0111 | 027 | 23  | 17  | ETB   | ^W                      |               | Bloco de fim de transmissão (inglês End of Transmission Block)         |
| 0001 1000 | 030 | 24  | 18  | CAN   | ^X                      |               | Cancelar (inglês Cancel)                                               |
| 0001 1001 | 031 | 25  | 19  | EM    | ^Y                      |               | Fim de mídia; fim do meio (inglês End of Medium)                       |
| 0001 1010 | 032 | 26  | 1A  | SUB   | ^Z                      |               | Substituir (inglês Substitute)                                         |
| 0001 1011 | 033 | 27  | 1B  | ESC   | ^[                      |               | Escapar (inglês Escape)                                                |
| 0001 1100 | 034 | 28  | 1C  | FS    | ^\                      |               | Separador de arquivos (inglês File Separator)                          |
| 0001 1101 | 035 | 29  | 1D  | GS    | ^]                      | \e            | Separador de grupos (inglês Group Separator)                           |
| 0001 1110 | 036 | 30  | 1E  | RS    | ^^                      |               | Separador de registos (inglês Record Separator)                        |
| 0001 1111 | 037 | 31  | 1F  | US    | ^_                      |               | Separador de unidades (inglês Unit Separator)                          |
| 0111 1111 | 177 | 127 | 7F  | DEL   | ^?                      |               | Deletar (inglês Delete)                                                |

Elementos gráficos (Imprimíveis)

| Bin       | Oct | Dec | Hex | Sinal    | Bin       | Oct | Dec | Hex | Sinal | Bin       | Oct | Dec | Hex | Sinal |
|-----------|-----|-----|-----|----------|-----------|-----|-----|-----|-------|-----------|-----|-----|-----|-------|
| 0010 0000 | 040 | 32  | 20  | (espaço) | 0100 0000 | 100 | 64  | 40  | @     | 0110 0000 | 140 | 96  | 60  | `     |
| 0010 0001 | 041 | 33  | 21  | !        | 0100 0001 | 101 | 65  | 41  | A     | 0110 0001 | 141 | 97  | 61  | a     |
| 0010 0010 | 042 | 34  | 22  | "        | 0100 0010 | 102 | 66  | 42  | B     | 0110 0010 | 142 | 98  | 62  | b     |
| 0010 0011 | 043 | 35  | 23  | #        | 0100 0011 | 103 | 67  | 43  | C     | 0110 0011 | 143 | 99  | 63  | c     |
| 0010 0100 | 044 | 36  | 24  | \$       | 0100 0100 | 104 | 68  | 44  | D     | 0110 0100 | 144 | 100 | 64  | d     |
| 0010 0101 | 045 | 37  | 25  | %        | 0100 0101 | 105 | 69  | 45  | E     | 0110 0101 | 145 | 101 | 65  | e     |
| 0010 0110 | 046 | 38  | 26  | &        | 0100 0110 | 106 | 70  | 46  | F     | 0110 0110 | 146 | 102 | 66  | f     |
| 0010 0111 | 047 | 39  | 27  | '        | 0100 0111 | 107 | 71  | 47  | G     | 0110 0111 | 147 | 103 | 67  | g     |
| 0010 1000 | 050 | 40  | 28  | (        | 0100 1000 | 110 | 72  | 48  | H     | 0110 1000 | 150 | 104 | 68  | h     |
| 0010 1001 | 051 | 41  | 29  | )        | 0100 1001 | 111 | 73  | 49  | I     | 0110 1001 | 151 | 105 | 69  | i     |
| 0010 1010 | 052 | 42  | 2A  | *        | 0100 1010 | 112 | 74  | 4A  | J     | 0110 1010 | 152 | 106 | 6A  | j     |
| 0010 1011 | 053 | 43  | 2B  | +        | 0100 1011 | 113 | 75  | 4B  | K     | 0110 1011 | 153 | 107 | 6B  | k     |
| 0010 1100 | 054 | 44  | 2C  | ,        | 0100 1100 | 114 | 76  | 4C  | L     | 0110 1100 | 154 | 108 | 6C  | l     |
| 0010 1101 | 055 | 45  | 2D  | -        | 0100 1101 | 115 | 77  | 4D  | M     | 0110 1101 | 155 | 109 | 6D  | m     |
| 0010 1110 | 056 | 46  | 2E  | .        | 0100 1110 | 116 | 78  | 4E  | N     | 0110 1110 | 156 | 110 | 6E  | n     |
| 0010 1111 | 057 | 47  | 2F  | /        | 0100 1111 | 117 | 79  | 4F  | O     | 0110 1111 | 157 | 111 | 6F  | o     |
| 0011 0000 | 060 | 48  | 30  | 0        | 0101 0000 | 120 | 80  | 50  | P     | 0111 0000 | 160 | 112 | 70  | p     |
| 0011 0001 | 061 | 49  | 31  | 1        | 0101 0001 | 121 | 81  | 51  | Q     | 0111 0001 | 161 | 113 | 71  | q     |
| 0011 0010 | 062 | 50  | 32  | 2        | 0101 0010 | 122 | 82  | 52  | R     | 0111 0010 | 162 | 114 | 72  | r     |
| 0011 0011 | 063 | 51  | 33  | 3        | 0101 0011 | 123 | 83  | 53  | S     | 0111 0011 | 163 | 115 | 73  | s     |
| 0011 0100 | 064 | 52  | 34  | 4        | 0101 0100 | 124 | 84  | 54  | T     | 0111 0100 | 164 | 116 | 74  | t     |
| 0011 0101 | 065 | 53  | 35  | 5        | 0101 0101 | 125 | 85  | 55  | U     | 0111 0101 | 165 | 117 | 75  | u     |
| 0011 0110 | 066 | 54  | 36  | 6        | 0101 0110 | 126 | 86  | 56  | V     | 0111 0110 | 166 | 118 | 76  | v     |
| 0011 0111 | 067 | 55  | 37  | 7        | 0101 0111 | 127 | 87  | 57  | W     | 0111 0111 | 167 | 119 | 77  | w     |
| 0011 1000 | 070 | 56  | 38  | 8        | 0101 1000 | 130 | 88  | 58  | X     | 0111 1000 | 170 | 120 | 78  | x     |
| 0011 1001 | 071 | 57  | 39  | 9        | 0101 1001 | 131 | 89  | 59  | Y     | 0111 1001 | 171 | 121 | 79  | y     |
| 0011 1010 | 072 | 58  | 3A  | :        | 0101 1010 | 132 | 90  | 5A  | Z     | 0111 1010 | 172 | 122 | 7A  | z     |
| 0011 1011 | 073 | 59  | 3B  | ;        | 0101 1011 | 133 | 91  | 5B  | [     | 0111 1011 | 173 | 123 | 7B  | [     |
| 0011 1100 | 074 | 60  | 3C  | <        | 0101 1100 | 134 | 92  | 5C  | \     | 0111 1100 | 174 | 124 | 7C  | ]     |
| 0011 1101 | 075 | 61  | 3D  | =        | 0101 1101 | 135 | 93  | 5D  | ]     | 0111 1101 | 175 | 125 | 7D  | ^     |
| 0011 1110 | 076 | 62  | 3E  | >        | 0101 1110 | 136 | 94  | 5E  | ^     | 0111 1110 | 176 | 126 | 7E  | _     |
| 0011 1111 | 077 | 63  | 3F  | ?        | 0101 1111 | 137 | 95  | 5F  | _     |           |     |     |     |       |

Fonte:

Wikipedia, 2020, acessível em:

<https://pt.wikipedia.org/wiki/ASCII>